

# LÖSUNGEN

if / else / else if ..... 2

    Aufgabe 1: ..... 2

    Aufgabe 2: ..... 2

    Aufgabe 3: ..... 2

    Aufgabe 4: ..... 2

    Aufgabe 5: ..... 3

    Aufgabe 6: ..... 3

    Aufgabe 7: ..... 3

    Aufgabe 8: ..... 4

    Aufgabe 9: ..... 4

    Aufgabe 10: ..... 5

switch ..... 8

    Aufgabe 1: ..... 8

    Aufgabe 2: ..... 8

    Aufgabe 3: ..... 9

Schleifen ..... 10

    Aufgabe 1: ..... 10

    Aufgabe 2: ..... 10

    Aufgabe 3: ..... 10

    Aufgabe 4: ..... 10

    Aufgabe 5: ..... 10

    Aufgabe 6: ..... 10

    Aufgabe 7: ..... 11

    Aufgabe 8: ..... 11

    Aufgabe 9: ..... 11

    Aufgabe 10: ..... 11

    Aufgabe 11: ..... 12

    Aufgabe 12: ..... 12

    Aufgabe 13: ..... 13

    Aufgabe 14: ..... 13

    Aufgabe 15: ..... 13

Array ..... 14

    Aufgabe 1: ..... 14

## if / else / else if

### Aufgabe 1:

```
public void Aufgabe1(int zahl)
{
    if (zahl < 0) {
        System.out.println("Negative Zahl");
    }
    else if (zahl <= 10) {
        System.out.println("Die Zahl ist kleiner gleich 10");
    }
    else if (zahl < 100) {
        System.out.println("Die Zahl ist zweistellig");
    }
    else if (zahl == 100) {
        System.out.println("Die Zahl ist 100");
    }
    else {
        System.out.println("101 und mehr");
    }
}
```

### Aufgabe 2:

```
public void Aufgabe2(boolean a, boolean b)
{
    if (!a && !b) {
        System.out.println("falsch");
    }
    else if (a || b) {
        System.out.println("wahr");
    }
    else {
        System.out.println("wahr");
    }
}
```

### Aufgabe 3:

```
public void Aufgabe3(boolean a, boolean b)
{
    if (a^b) {
        System.out.println("wahr");
    }
    else {
        System.out.println("falsch");
    }
}
```

### Aufgabe 4:

```
public void Aufgabe4(int a, int b)
{
    if ((a < 0) && (b < 0)) {
        System.out.println("Beide Zahlen sind negativ");
    }
    else if ((a > 0) && (b > 0)) {
        System.out.println("Beide Zahlen sind positiv");
    }
    else {
        System.out.println("Eine der Zahlen ist positiv");
    }
}
```

**Aufgabe 5:**

```
public void Aufgabe5 (int a, int b)
{
    int c;
    c = a*b;

    if (c < 0) {
        System.out.println("Das Ergebnis ist negativ");
    }
    else {
        System.out.println("Das Ergebnis ist positiv");
    }
}
```

**Aufgabe 6:**

```
public boolean Aufgabe6 (int wetter, int temp)
{
    if (((temp > 50) || (temp < -50)) || ((wetter < 0) || (wetter > 4))) {
        return false;
    }

    if (wetter == 1) {
        return true;
    }
    else if ((wetter == 3) && (temp > 25)) {
        return true;
    }
    else {
        return false;
    }
}
```

**Aufgabe 7:**

```
//Param1 = Der zu bezahlende Betrag
//Param2 = Aktuelles Vermögen
//Param3 = Geld das durch Häuser verkauft eingenommen werden könnte
public void Aufgabe7 (int zuBezahlenderBetrag, int vermoegen, int geldHaeuserVerkauf)
{
    if (vermoegen >= zuBezahlenderBetrag){
        System.out.println("bezahlen");
    }
    else if ((geldHaeuserVerkauf + vermoegen) >= zuBezahlenderBetrag) {
        System.out.println("Betrag zu bezahlen durch Häuser Verkauf " + (zuBezahlenderBetrag -
vermoegen));
    }
    else {
        System.out.println("Game Over");
    }
}
```

**Aufgabe 8:**

```
public boolean Aufgabe8(int wochentag, int uhrzeit, boolean pruefungsvorbereitungszeit)
{
    if (((wochentag > 0) && (wochentag <= 4)) && (uhrzeit >= 6 )){
        return true;
    }
    else if ((wochentag == 5) && (uhrzeit >= 7)) {
        return true;
    }
    else if (((wochentag == 6) && (wochentag == 7))) {
        if ( pruefungsvorbereitungszeit ) {
            if (uhrzeit >= 9 ) {
                return true;
            }
            else if (uhrzeit >= 10) {
                return true;
            }
            else {
                return false;
            }
        }
        else {
            return false;
        }
    }
    else {
        return false;
    }
}
```

**Aufgabe 9:**

```
// 1 = Schere, 2 = Stein, 3 = Papier
public void Aufgabe9(int spieler1, int spieler2)
{
    if (spieler1 == spieler2) {
        System.out.println("Gleichstand. Niemand gewinnt.");
    }
    else if (((spieler1 == 1) && (spieler2 == 2)) || ((spieler1 == 2) && (spieler2 == 3)) ||
((spieler1 == 3) && (spieler2 == 1))) {
        System.out.println("Spieler2 gewinnt.");
    }
    else {
        System.out.println("Spieler1 gewinnt.");
    }
}
```

**Aufgabe 10:**

```

import java.util.Random;
/**
 * Schere Stein Papier für drei Spieler
 * Ich erhebe keinen Anspruch darauf, dass mein Code effizient & sauber ist...
 * Ein paar Ideen für Verbesserungen hätte ich, aber bis jetzt hatte ich noch keine zeit (und Lust)
 dies anzupassen ;-))
 *
 * @author Felix Rohrer
 * @version 1.3
 */
public class SchereSteinPapier
{
    // welche wahl gibt es für die Spieler bei Schere-Stein-Papier
    private String[] wahlSSP = {"Schere", "Stein", "Papier"};
    // Rundenzähler
    private int runde;
    // Random Generator
    private Random generator;

    /**
     * Constructor for objects of class SchereSteinPapier
     */
    public SchereSteinPapier()
    {
        // init Random
        generator = new Random();
    }

    /**
     * Play...
     */
    public void play()
    {
        // Rundenzähler auf 0 setzen, noch hat niemand gewonnen
        runde = 0;
        boolean jemandHatGewonnen = false;

        // Spieler und Gewinner Variablen initialisieren
        int wahlPlayer1, wahlPlayer2, wahlPlayer3, gewinner;

        // solange spielen bis jemand gewonnen hat
        do {
            // Runde ausgeben
            runde++;
            System.out.println("Runde: " + runde);

            // Die "wahl" der Spieler per zufall bestimmen
            wahlPlayer1 = generator.nextInt(3) + 1;
            wahlPlayer2 = generator.nextInt(3) + 1;
            wahlPlayer3 = generator.nextInt(3) + 1;
            System.out.println("Spieler 1: " + wahlSSP[wahlPlayer1 - 1]);
            System.out.println("Spieler 2: " + wahlSSP[wahlPlayer2 - 1]);
            System.out.println("Spieler 3: " + wahlSSP[wahlPlayer3 - 1]);

            // wer gewinnt?
            gewinner = schereSteinPapier(wahlPlayer1, wahlPlayer2, wahlPlayer3);
            // Gewinner überprüfen & ausgeben
            if (gewinner != -1) {
                // wir haben einen gewinner!
                jemandHatGewonnen = true;
                System.out.println("*****");
                System.out.println("*   Spieler " + gewinner + " hat gewonnen!   *");
                System.out.println("*****\n\n");
            }
            else {
                //niemand hat gewonnen, die nächste Runde beginnt
                System.out.println("Niemand hat gewonnen...\n");
            }
        } while (!jemandHatGewonnen);
    } // while (!jemandHatGewonnen);
}

```

```

/**
 * Schere-Stein-Papier mit 3 Spielern.
 * @param p1 Spieler1, mögliche Werte: (1 = Schere, 2 = Stein, 3 = Papier)
 * @param p2 Spieler2, mögliche Werte: (1 = Schere, 2 = Stein, 3 = Papier)
 * @param p3 Spieler3, mögliche Werte: (1 = Schere, 2 = Stein, 3 = Papier)
 * @return welcher Spieler gewonnen hat. (-1 = niemand, 1 = Spieler1, 2 = Spieler2, 3 = Spieler
3)
 */
private int schereSteinPapier(int p1, int p2, int p3)
{
    // init retourwert
    int returnWinner = -1;

    // alle sind gleich oder alle sind unterschiedlich ==> niemand kann gewinnen
    if (((p1 == p2) && (p2 == p3) && (p1 == p3))
        || ((p1 != p2) && (p2 != p3) && (p1 != p3))) {
        // niemand hat gewonnen
    }
    else {
        // mindestens eine person gewinnt

        // einzelne Spieler "vergleichen"
        int player1VsPlayer2 = werGewinntSchereSteinPapier(p1, p2);
        int player1VsPlayer3 = werGewinntSchereSteinPapier(p1, p3);
        int player2VsPlayer3 = werGewinntSchereSteinPapier(p2, p3);

        boolean player1Couldwin = true;
        boolean player2Couldwin = true;
        boolean player3Couldwin = true;

        // player 1 vs. player 2
        switch (player1VsPlayer2) {
            // player 1 gewinnt gegen Player 2
            case 1:
                player2Couldwin = false; // Player 2 kann nicht mehr gewinnen
                // player 1 vs. player 3
                switch (player1VsPlayer3) {
                    case 1: returnWinner = 1; break; // player 1 gewinnt
                    case 2: returnWinner = 3; break; // player 3 gewinnt
                }
                break;

            // player 2 gewinnt gegen Player 1
            case 2:
                player1Couldwin = false; // Player 1 kann nicht mehr gewinnen
                // player 2 vs. player 3
                switch (player2VsPlayer3) {
                    case 1: returnWinner = 2; break; // player 2 gewinnt
                    case 2: returnWinner = 3; break; // player 3 gewinnt
                }
                break;

            // player 1 und 2 unentschieden - ggf. gewinnt player 3?
            case -1:
                if (player1VsPlayer3 == 2) {
                    returnWinner = 3; // player 3 gewinnt
                }
                else {
                    player3Couldwin = false; // Player 3 kann nicht mehr gewinnen
                }
                break;
        }

        if (returnWinner == -1) {
            do {
                // es gab noch keinen gewinner - 2 spieler sind unentschieden
                System.out.println("Niemand hat gewonnen...\n");

                // wer muss nochmals spielen?
                if (player1Couldwin && player2Couldwin) {
                    //player 1 und player 2 gehen in die nächste Runde
                    returnWinner = schereSteinPapier2Player(1, 2);
                }
                else if (player1Couldwin && player3Couldwin) {
                    //player 1 und player 3 gehen in die nächste Runde

```

```

        returnWinner = schereSteinPapier2Player(1, 3);
        if (returnWinner == 2) { returnWinner = 3; }
    }
    else {
        //player 2 und player 3 gehen in die nächste Runde
        returnWinner = schereSteinPapier2Player(2, 3);
        if (returnWinner == 1) { returnWinner = 2; }
        else if (returnWinner == 2) { returnWinner = 3; }
    }
} while (returnWinner == -1);
}
} // unentschieden

// Gewinner (oder Unentschieden) zurück geben
return returnWinner;
}

/**
 * Schere-Stein-Papier mit 2 Spielern.
 * @return welcher Spieler gewonnen hat. (-1 = niemand, 1 = Spieler1, 2 = Spieler2)
 */
private int schereSteinPapier2Player(int player1, int player2)
{
    // Spieler und Gewinner Variablen initialisieren
    int wahlPlayer1, wahlPlayer2;

    // Runde ausgeben
    System.out.println("Neue Runde mit Spieler " + player1 + " und Spieler " + player2 +
"...");
    runde++;
    System.out.println("Runde: " + runde);

    // Die "wahl" der Spieler per zufall bestimmen
    wahlPlayer1 = generator.nextInt(3) + 1;
    wahlPlayer2 = generator.nextInt(3) + 1;
    System.out.println("Spieler " + player1 + ": " + wahlSSP[wahlPlayer1 - 1]);
    System.out.println("Spieler " + player2 + ": " + wahlSSP[wahlPlayer2 - 1]);

    // player 1 vs. player 2
    return werGewinntSchereSteinPapier(wahlPlayer1, wahlPlayer2);
}

/**
 * wer Gewinnt bei Schere-Stein-Papier wenn zwei Spieler gegeneinander spielen.
 * @param p1 Spieler1, mögliche Werte: (1 = Schere, 2 = Stein, 3 = Papier)
 * @param p2 Spieler2, mögliche Werte: (1 = Schere, 2 = Stein, 3 = Papier)
 * @return Zahl welcher Spieler gewonnen hat (-1 = niemand gewinnt, 1 = Spieler1, 2 = Spieler
2)
 */
private int werGewinntSchereSteinPapier(int p1, int p2)
{
    // init retourwert
    int returnWinner = -1;

    // gleichstand?
    if (p1 != p2) {
        // player 1 vs. player 2
        if (((p1 == 2) && (p2 == 1)) || ((p1 == 3) && (p2 == 2)) || ((p1 == 1) && (p2 == 3))) {
            // player 1 wins
            returnWinner = 1;
        }
        else {
            // player 2 wins
            returnWinner = 2;
        }
    }
    // else wird nicht gebraucht, da der Rückgabewert per Default -1 ist.

    // Gewinner (oder Unentschieden) zurück geben
    return returnWinner;
}
}
}

```

## switch

### Aufgabe 1:

```
public void aufgabe1(int monat) {
    switch (monat) {
        case 1:
            System.out.println("Januar");
            break;
        case 2:
            System.out.println("Februar");
            break;
        case 3:
            System.out.println("März");
            break;
        case 4:
            System.out.println("April");
            break;
        case 5:
            System.out.println("Mai");
            break;
        case 6:
            System.out.println("Juni");
            break;
        case 7:
            System.out.println("Juli");
            break;
        case 8:
            System.out.println("August");
            break;
        case 9:
            System.out.println("September");
            break;
        case 10:
            System.out.println("Oktober");
            break;
        case 11:
            System.out.println("November");
            break;
        case 12:
            System.out.println("Dezember");
            break;
        default:
            System.out.println("Ungültiger Monat");
    }
}
```

### Aufgabe 2:

```
public void aufgabe2(int hour) {
    switch(hour) {
        case 5: case 6: case 7: case 8: case 9: case 10: case 11:
            System.out.println("Guten Morgen");
            break;
        case 12:
            System.out.println("Guten Mittag");
            break;
        case 13: case 14: case 15: case 16: case 17:
            System.out.println("Guten Nachmittag");
            break;
        case 18: case 19: case 20:
            System.out.println("Guten Abend");
            break;
        case 21: case 22: case 23: case 24: case 1: case 2: case 3: case 4:
            System.out.println("Gute Nacht");
            break;
        default:
            System.out.println("keine gültige Stunde");
            break;
    }
}
```



**Aufgabe 3:**

```
public void aufgabe3(int month, int year) {

    int numberOfDays = 0;
    int years = year;

    switch(month) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            numberOfDays = 31;
            break;
        case 4: case 6: case 9: case 11:
            numberOfDays = 30;
            break;
        case 2:
            if((year % 4 == 0) && !(year % 100 == 0) || (year % 400 == 0)) {
                numberOfDays = 29;
            }
            else {
                numberOfDays = 28;
            }
    }
    System.out.println("Number of Days: " + numberOfDays);
}
```

## Schleifen

### Aufgabe 1:

for / while / do-while

### Aufgabe 2:

do-while (while wäre auch möglich)

### Aufgabe 3:

do-while (mindestens einen Durchlauf)

### Aufgabe 4:

```
do {  
    // input einlesen  
} while (input >=0 && input < 10);
```

### Aufgabe 5:

```
public void Aufgabe5()  
{  
    int i = 1;  
    while(i<=10)  
    {  
        System.out.println(i);  
        i++;  
    }  
  
    i = 1;  
    do {  
        System.out.println(i);  
        i++;  
    } while (i <= 10);  
}
```

### Aufgabe 6:

```
public void Aufgabe6a()  
{  
    for(int i=20; i>=10; i--) {  
        System.out.println(i);  
    }  
}  
  
public void Aufgabe6b()  
{  
    int i = 20;  
    while(i>=10)  
    {  
        System.out.println(i);  
        i--;  
    }  
}
```

**Aufgabe 7:**

```
public void Aufgabe7()
{
    for(int i=0; i<=50; i = i + 2) {
        System.out.println(i);
    }
}
```

**Aufgabe 8:**

```
public void Aufgabe8a()
{
    for(int i=0; i<=100; i += 3) {
        System.out.println(i);
    }
}

public void Aufgabe8b()
{
    int i = 0;
    while (i <= 100) {
        System.out.println(i);
        i += 3; // entspricht i = i + 3
    }
}
```

**Aufgabe 9:**

```
public void Aufgabe9()
{
    int i = 0;
    int j = 1;
    while (i <= 500) {
        System.out.println(i);
        i = i + j;
        j++;
    }
}
```

**Aufgabe 10:**

```
public void aufgabe10(int a, int b)
{
    if (a > b) {

        // a und b vertauschen
        int c;
        c = a; // a zwischenspeichern
        a = b; // b in a speichern
        b = c; // a (resp. temporär c) in b speichern

    }
    for (int i = a; i <= b; i++)
    {
        System.out.println(i);
    }
}
```

**Aufgabe 11:**

```
public void aufgabe11()
{
    int i = 0;
    int j = 100;

    while(i<j)
    {
        System.out.println(i + " ist noch kleiner als " + j);
        i += 2;
        j -= 3;
    }
}
```

**Aufgabe 12:**

```
public void aufgabe12(int zeile, int spalte)
{
    for(int i = 0; i < zeile; i++)
    {
        for(int j = 0; j < spalte; j++)
        {
            System.out.print("#");
        }

        System.out.println();
    }
}

public void aufgabe12b(int zeile, int spalte)
{
    int i = 0;

    while(i<zeile)
    {
        System.out.println();
        i++;

        int j = 0;
        while(j<spalte)
        {
            System.out.print("#");
            j++;
        }
    }
}

public void aufgabe12c(int zeile, int spalte)
{
    int i = 0;
    do {
        System.out.println();
        i++;

        int j = 0;
        do {
            System.out.print("#");
            j++;
        } while (j < spalte);
    } while (i < zeile);
}
```

### Aufgabe 13:

```
public void aufgabe13()
{
    for(int i = 1; i<10; i++)
    {
        for(int j = 1; j<=i; j++)
        {
            System.out.print(j);
        }

        System.out.println();
    }
}

public void aufgabe13b()
{
    int i = 1;
    while(i<10)
    {
        System.out.println();
        int j = 1;
        while(j<=i)
        {
            System.out.print(j);
            j++;
        }

        i++;
    }
}

public void aufgabe13c()
{
    int i = 1;
    do
    {
        System.out.println();
        int j = 1;
        do
        {
            System.out.print(j);
            j++;
        } while ( j<=i);

        i++;
    }while (i < 10);
}
```

### Aufgabe 14:

```
+++++.
+++++.
+++++.
+++++.
+++++.
+++++.
++++.
+++.
++.
+.
```

### Aufgabe 15:

```
-----*-----
-----***-----
-----*****-----
-----*****-----
-----*****-----
-----*****-----
-----*****-----
*****
```

## Array

### Aufgabe 1:

```
public void aufgabe1(int month) {
    String[] monthList = {"Januar", "Februar", "März", "April", "Mai", "Juni", "Juli",
"August", "September", "Oktober", "November", "Dezember"};
    if (month >= 1 && month <= 12) {
        System.out.println(monthList[month - 1]);
    }
    else {
        System.out.println(monthList[month - 1]);
    }
}
```