

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class addieren {  
  
    /*  
    * Aufgabe 1:  
    * Implementieren Sie eine Methode, die alle Zahlen von 1 bis zu einer  
    * definierten Zahl n addiert. Beispiel addiere(3) addiert 1 + 2 + 3 = 6  
    */  
  
    public int addieren(int n) {  
        int ergebnis = 0;  
        while (n>0)  
            ergebnis += n--;  
        return ergebnis;  
    }  
  
    public static void main(String args[]) {  
        addieren a = new addieren();  
        int ergebnis1 = a.addieren(1);  
        int ergebnis3 = a.addieren(3);  
        int ergebnis6 = a.addieren(6);  
        int ergebnis100 = a.addieren(100);  
        System.out.println("addieren(1) hat "+ergebnis1+" berechnet, richtig ist 1");  
        System.out.println("addieren(3) hat "+ergebnis3+" berechnet, richtig ist 6");  
        System.out.println("addieren(6) hat "+ergebnis6+" berechnet, richtig ist 21");  
        System.out.println("addieren(100) hat "+ergebnis100+" berechnet, richtig ist 5050");  
    }  
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class zaehlen {

    /*
     * Aufgabe 2:
     * Implementieren Sie eine Methode, die als Eingabe einen beliebigen String s
     * und ein Zeichen z erhält. Aufgabe der Methode ist es, zu zählen, wie oft das
     * Zeichen z im String s enthalten ist. Beim Vergleichen ist auf Groß-
     * /Kleinschreibung zu achten.
     */

    public int zaehlen(String s, char z) {
        int ergebnis = 0;
        for (int i=0; i<s.length(); i++)
            if (s.charAt(i) == z)
                ergebnis++;
        return ergebnis;
    }

    public static void main(String args[]) {
        zaehlen a = new zaehlen();
        int ergebnis1 = a.zaehlen("Das Leben ist schoen", 'e');
        int ergebnis2 = a.zaehlen("Das Leben ist schoen", 'l');
        int ergebnis3 = a.zaehlen("Das Leben ist schoen", 'u');
        System.out.println("Der String lautet: Das Leben ist schoen");
        System.out.println("Ergebnis der Methode zaehlen fuer e ist "+ergebnis1+", richtig ist 3");
        System.out.println("Ergebnis der Methode zaehlen fuer l ist "+ergebnis2+", richtig ist 0");
        System.out.println("Ergebnis der Methode zaehlen fuer u ist "+ergebnis3+", richtig ist 0");
    }
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class suchen{

    /*
    * Aufgabe 3:
    * Implementieren Sie eine Methode, die als Eingabe zwei Strings s1 und s2
    * enthält. Sie dürfen annehmen, dass s1 länger ist als s2. Die Methode soll
    * überprüfen, ob der String s2 im String s1 enthalten ist. Hierbei ist auf Groß-
    * /Kleinschreibung zu achten.
    * Beispiel: s1="Das Leben ist schoen", s2 ="schoen" würde wahr ergeben,
    * s2="Schoen" würde falsch ergeben.
    */

    public boolean suche(String s1, String s2) {
        boolean ergebnis = false;
        for (int i=0; i<(s1.length()-s2.length()); i++) {
            if (s1.charAt(i) == s2.charAt(0)) {
                boolean gleich = true;
                for (int u=0; u<s2.length(); u++)
                    if (s1.charAt(i+u) != s2.charAt(u))
                        gleich = false;
                if (gleich == true)
                    return true;
            }
        }
        return ergebnis;
    }

    public static void main(String args[]) {
        suchen a = new suchen();
        boolean s1 = a.suche("Das Leben ist schoen","ist");
        boolean s2 = a.suche("Das Leben ist schoen","Leben");
        boolean s3 = a.suche("Das Leben ist schoen","leben");
        System.out.println("Der String lautet: Das Leben ist schoen");
        System.out.println("Suche: ist Ergebnis:"+s1+" richtig ist true");
        System.out.println("Suche: Leben Ergebnis:"+s2+" richtig ist true");
        System.out.println("Suche: leben Ergebnis:"+s3+" richtig ist false");
    }
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class zaehlen2 {

    /*
    * Aufgabe 4:
    * Implementieren Sie eine Methode, die als Eingabe einen beliebigen String
    * s1 akzeptiert und die verschiedenen Zeichen in diesem String zählt (wieder
    * soll auf Groß-/Kleinschreibung geachtet werden. Die Methode soll in der
    * Ausgabe dann alle Zeichen und deren Anzahl ausgeben, die mindestens
    * einmal vorhanden sind. Zeichen, die nicht vorhanden sind, sollen auch nicht
    * ausgegeben werden.
    * Beispiel: s1="lHallo !", Ausgabe: "l":1 "H":1 "a":1 "l":2 "o":1 "!":1 " ":1
    */

    public void zaehleZeichen(String s) {
        char zeichen;
        int anzahl;
        boolean doppelt;
        System.out.println("Der String \"" + s + "\" enthaelt die folgenden Zeichen:");
        for (int i=0; i<s.length(); i++) {
            anzahl = 0;
            doppelt = false;
            zeichen = s.charAt(i);
            for (int u=0; u<i; u++)
                if (s.charAt(u) == zeichen)
                    doppelt = true;
            if (!doppelt) {
                for (int u=i; u<s.length(); u++)
                    if (s.charAt(u) == zeichen)
                        anzahl++;
                System.out.print("\"" + zeichen + "\": " + anzahl + " ");
            }
        }
    }

    public static void main(String args[]) {
        zaehlen2 a = new zaehlen2();
        System.out.println("Erwartete Ausgabe:");
        System.out.println("Der String \"Das Leben ist schoen!\" enthaelt die folgenden Zeichen:\n'D': 1 'a': 1 's': 3 ' ': 3 'L': 1 'e': 3 'b': 1 '\n': 2 'i': 1 't': 1 'c': 1 'h': 1 'o': 1 '!': 1\nDer String \"Programmierung\" enthaelt die folgenden Zeichen:\n'P': 1 'r': 3 'o': 1 'g': 2 'a': 1 'm': 2 'i': 1 'e': 1 'u': 1 '\n': 1");
        System.out.println();
        System.out.println("Ihre Ausgabe:");
        a.zaehleZeichen("Das Leben ist schoen!");
        System.out.println();
        a.zaehleZeichen("Programmierung");
    }
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class caesar {

    /*
    * Aufgabe 5:
    * Caesar-Chiffre nennt man ein Verschlüsselungsverfahren, das die
    * Buchstaben eines Wortes um eine definierte Anzahl an Zeichen verschiebt.
    * Beispiel: Wort="haus", Verschiebung = 1, chiffriertes Wort: "ibvt"
    * Implementieren Sie eine Methode, die als Eingabe einen String s1 und eine
    * Zahl i vom Typ integer (aber nur positive) akzeptiert und die Caesar-Chiffre
    * von s1 ausgibt. Vereinfachend dürfen Sie annehmen, dass der String s1 nur
    * aus Kleinbuchstaben von a bis z besteht (kein ö,ä,ü,ß). Zwei Tipps: Der
    * Nachfolgebuchstabe von z ist a und man kann String und Character mit dem
    * + Zeichen addieren. Zum Beispiel ist „Hau“ + ‚s‘ = „Haus“.
    */

    public String chiffre(String s, int v) {
        String ergebnis = "";
        int zahl;
        for (int i=0; i<s.length(); i++) {
            zahl = (int)s.charAt(i) + v;
            if (zahl > 122)
                zahl -= 26;
            ergebnis += (char)zahl;
        }
        return ergebnis;
    }

    public static void main(String args[]) {
        caesar c = new caesar();
        System.out.println("Casesar-Chiffre von 'haus' mit 1 ist "+c.chiffre("haus",1)+" richtig ist:
ibvt");
        System.out.println("Casesar-Chiffre von 'bauzaun' mit 8 ist "+c.chiffre("bauzaun",8)+" richtig
ist: jichicv");
        System.out.println("Casesar-Chiffre von 'hundert' mit 10 ist "+c.chiffre("hundert",10)+" rich-
tig ist: rexnobd");
        System.out.println("Casesar-Chiffre von 'rexnobd' mit 16 ist "+c.chiffre("rexnobd",16)+" rich-
tig ist: hundert");
    }
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class fibonacci {

    /*
     * Aufgabe 6:
     * Die Fibonacci-Reihe über die natürlichen Zahlen ist wie folgt definiert:
     * f(1)=1 f(2)=1 f(n)=f(n-1) + f(n-2), mit n>2
     * Zum Beispiel: f(4) = f(3) + f(2) = f(1) + f(1) + f(1) = 1 + 1 + 1 = 3
     * Implementieren Sie eine Methode, die die Fibonacci-Reihe der Integerzahl n
     * ausgibt. Einmal rekursiv und einmal iterativ. Falls Sie nicht wissen, was
     * rekursiv oder iterativ ist, versuchen einfach irgendwie die Fibonacci-Reihe zu
     * implementieren.
     */

    // Fibonacci rekursiv
    public int fibonaccirek(int f) {
        if (f==2)
            return 1;
        else if (f == 1)
            return 1;
        else
            return fibonaccirek(f-1) + fibonaccirek(f-2);
    }

    // Fibonacci iterativ
    public int fibonacciiter(int f) {
        int f1 = 1;
        int f2 = 1;
        int tmp;
        for (int i=0; i<f-2; i++) {
            tmp = f1;
            f1 += f2;
            f2 = tmp;
        }
        return f1;
    }

    public static void main(String args[]) {
        fibonacci f = new fibonacci();
        System.out.println("fib(5) = "+f.fibonacciiter(5)+" (iterativ)");
        System.out.println("fib(5) = "+f.fibonaccirek(5)+" (rekursiv)");
        System.out.println("Richtig ist fib(5) = 5");
        System.out.println("fib(15) = "+f.fibonacciiter(15)+" (iterativ)");
        System.out.println("fib(15) = "+f.fibonaccirek(15)+" (rekursiv)");
        System.out.println("Richtig ist fib(15) = 610");
        System.out.println("fib(40) = "+f.fibonacciiter(40)+" (iterativ)");
        System.out.println("fib(40) = "+f.fibonaccirek(40)+" (rekursiv)");
        System.out.println("Richtig ist fib(40) = 102334155");
        System.out.println("Warum ist der rekursive Algo nur so langsam...?");
    }
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
class eratosthenes {  
  
    /*  
    * Aufgabe 7:  
    * Das folgende Verfahren zur Primzahlen-Gewinnung ist als "Sieb des  
    * Eratosthenes" bekannt: Streicht man aus der Menge natürlicher Zahlen von  
    * 2 bis n, beginnend mit 2, nacheinander alle Vielfachen der jeweils noch  
    * verbleibenden Zahlen heraus, enthält die Restmenge nur noch Primzahlen.  
    * Ein Beispiel für die Zahlen 2 bis 15:  
    * 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
    * bearbeite 2  
    * 2 3 5 7 9 11 13 15  
    * bearbeite 3  
    * 2 3 5 7 11 13  
    * bearbeite 5, 7, 11, 13  
    * 2 3 5 7 11 13  
    * Implementieren Sie eine Methode, die anhand dieses Verfahrens die  
    * Primzahlen unter den ersten 50 bzw. 100 natürlichen Zahlen bestimmt und  
    * diese anschließend ausgibt.  
    */  
  
    public void sieb(int max) {  
        int[] a = new int[max];  
        for (int i=1; i<=max; i++)  
            a[i-1]=i;  
        for (int i=1; i<a.length; i++) {  
            if (a[i] > 0) {  
                for (int u=i+1; u<a.length; u++)  
                    if (a[u] % a[i] == 0)  
                        a[u] = -1;  
            }  
        }  
        for (int i=0; i<a.length; i++)  
            if (a[i] > 0)  
                System.out.print(a[i] + " ");  
    }  
  
    public static void main (String args[]) {  
        eratosthenes s = new eratosthenes();  
        System.out.println();  
        System.out.println("Sieb der ersten 50 Zahlen");  
        System.out.println("Erwartetes Ergebnis:");  
        System.out.println("1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47");  
        System.out.println("Ihre Loesung:");  
        s.sieb(50);  
        System.out.println();System.out.println();  
        System.out.println("Sieb der ersten 100 Zahlen");  
        System.out.println("Erwartetes Ergebnis:");  
        System.out.println("1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97");  
        System.out.println("Ihre Loesung:");  
        s.sieb(100);  
    }  
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class Matrix {  
  
    /*  
    * Aufgabe 8a:  
    * Gegeben sei eine zweidimensionale quadratische Matrix(Tabelle), die  
    * mit Hilfe eines Arrays realisiert ist. Die Felder der Matrix können nur  
    * Werte vom Datentyp Integer enthalten. Die Größe (Seitenlänge) kann  
    * beliebig gewählt werden.  
    * Im Programmcode ist eine Beispielmatrix mit der Seitenlänge 5 erzeugt  
    * worden, auf der die folgenden Aufgaben angewendet werden können.  
    * Beachten Sie: Matrixfelder die nicht explizit mit Werten gefüllt werden  
    * enthalten den Integer 0.  
    * - Implementieren Sie eine Methode, die die Summe jeder Spalte  
    * der Matrix ausgibt!  
    * - Implementieren Sie eine Methode die die Gesamtsumme der in  
    * der Matrix vorkommenden Werte ausgibt.  
    */  
  
    int [][] matrix;  
  
    public void erzeugematrix (int groesse) {  
        matrix = new int[groesse][groesse];  
        for (int i=0; i<groesse; i++)  
            for(int j=0; j<groesse; j++)  
                matrix[i][j]=0;  
    }  
  
    public void fuegein (int zahl, int positionX, int positionY) {  
        matrix[positionX][positionY]=zahl;  
    }  
  
    public void spaltensumme() {  
        int ergebnis;  
        for (int i=0; i<matrix.length; i++) {  
            ergebnis = 0;  
            for (int u=0; u<matrix[i].length; u++)  
                ergebnis += matrix[i][u];  
            System.out.println("Die Summe der Spalte " + i + " betraegt: " + ergebnis);  
        }  
    }  
  
    public void gesamtsumme() {  
        int ergebnis = 0;  
        for (int i=0; i<matrix.length; i++)  
            for (int u=0; u<matrix[i].length; u++)  
                ergebnis += matrix[i][u];  
        System.out.println("Die Gesamtsumme der Matrix betraegt: " + ergebnis);  
    }  
  
    public static void main (String args[]) {  
        Matrix m = new Matrix();  
        m.erzeugematrix (5);  
        m.fuegein (3,1,1);  
        m.fuegein (4,1,3);  
        m.fuegein (7,1,4);  
        m.fuegein (2,0,4);  
        m.fuegein (1,4,4);  
        m.fuegein (9,3,3);  
        m.fuegein (10,2,2);  
        m.fuegein (15,4,2);  
        m.fuegein (6,3,0);  
        m.fuegein (5,0,0);  
        System.out.println("Die Lösung sollte wie folgt aussehen:");  
        System.out.println("Die Summe der Spalte 0 betraegt: 7");  
        System.out.println("Die Summe der Spalte 1 betraegt: 14");  
        System.out.println("Die Summe der Spalte 2 betraegt: 10");  
        System.out.println("Die Summe der Spalte 3 betraegt: 15");  
        System.out.println("Die Summe der Spalte 4 betraegt: 16");  
        System.out.println("Die Gesamtsumme der Matrix betraegt: 62");  
        m.spaltensumme();  
        m.gesamtsumme();  
    }  
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
public class MatrixChar {

    /*
    * Aufgabe 8b:
    * Gegeben sei eine zweidimensionale quadratische Matrix(Tabelle), die
    * mit Hilfe von einem Array realisiert ist. Die Felder der Matrix können nur
    * Werte vom Datentyp Char enthalten, wir beschränken uns aber auf
    * Grossbuchstaben von A-Z. Die Größe (Seitenlänge) kann beliebig
    * gewählt werden.
    * Im Programmcode ist eine Beispielmatrix mit der Seitenlänge 9 erzeugt
    * worden, auf der die folgenden Aufgaben angewendet werden können.
    * Beachten Sie: Matrixfelder die nicht explizit mit Werten gefüllt werden
    * enthalten willkürlich gewählte Character.
    * - Implementieren Sie eine Methode, die nach folgenden Wörtern
    * Spalten- bzw. Zeilenweise in der Matrix sucht. Der
    * Rückgabewert der Methode soll true liefern falls das Wort in der
    * Matrix vorhanden ist, ist dies nicht der Fall wird false
    * zurückgegeben.
    * Suchen Sie nach folgenden Wörtern:
    * ESEL
    * EIMER
    * SSE
    * MENSA
    * QUAL
    * ARRAY
    * BRATEN
    */

    char [][] matrix;

    public void erzeugematrix (int groesse) {
        int puffer;
        matrix = new char[groesse][groesse];
        for (int i =0; i<groesse;i++)
            for(int j=0; j<groesse;j++) {
                puffer=(i+j)%26+65;
                matrix[i][j]=(char)puffer;
            }
    }

    public void fuegein (char chr, int positionY, int positionX) {
        matrix[positionX][positionY]=chr;
    }

    public static void main (String args[]){
        MatrixChar m = new MatrixChar();
        m.erzeugematrix (9);
        m.fuegein ('E',2,2);
        m.fuegein ('S',1,3);
        m.fuegein ('S',2,3);
        m.fuegein ('E',2,4);
        m.fuegein ('E',3,3);
        m.fuegein ('M',4,2);
        m.fuegein ('E',5,2);
        m.fuegein ('R',6,2);
        m.fuegein ('L',2,5);
        m.fuegein ('E',4,3);
        m.fuegein ('N',4,4);
        m.fuegein ('S',4,5);
        m.fuegein ('A',4,6);
        m.fuegein ('A',2,7);
        m.fuegein ('R',3,7);
        m.fuegein ('R',4,7);
        m.fuegein ('A',5,7);
        m.fuegein ('Y',6,7);
        m.fuegein ('Q',5,5);
        m.fuegein ('L',5,8);
        m.fuegein ('B',6,1);
        m.fuegein ('A',6,3);
        m.fuegein ('T',6,4);
        m.fuegein ('E',6,5);
        m.fuegein ('N',6,6);
        m.ausgabe();
        System.out.println("Die Lösung sollte wie folgt aussehen: ");
    }
}
```

Software Entwicklung & Programmierung - 0. Übungsblatt

```
System.out.println("Ist das Wort ESEL in der Spalte vorhanden: true");
System.out.println("Ist das Wort ESEL in der Zeile vorhanden: false");
System.out.println("Ist das Wort EIMER in der Spalte vorhanden: false");
System.out.println("Ist das Wort EIMER in der Zeile vorhanden: false");
System.out.println("Ist das Wort SSE in der Spalte vorhanden: false");
System.out.println("Ist das Wort SSE in der Zeile vorhanden: true");
System.out.println("Ist das Wort MENSA in der Spalte vorhanden: true");
System.out.println("Ist das Wort MENSA in der Zeile vorhanden : false");
System.out.println("Ist das Wort QUAL in der Spalte vorhanden: false");
System.out.println("Ist das Wort QUAL in der Zeile vorhanden: false");
System.out.println("Ist das Wort ARRAY in der Spalte vorhanden: false");
System.out.println("Ist das Wort ARRAY in der Zeile vorhanden: true");
System.out.println("Ist das Wort BRATEN in der Spalte vorhanden: true");
System.out.println("Ist das Wort BRATEN in der Zeile vorhanden: false");

System.out.println("\nMeine Lösung:");
System.out.println("Ist das Wort ESEL in der Spalte vorhanden: " + m.findeWortSP("ESEL"));
System.out.println("Ist das Wort ESEL in der Zeile vorhanden: " + m.findeWortZE("ESEL"));
System.out.println("Ist das Wort EIMER in der Spalte vorhanden: " + m.findeWortSP("EIMER"));
System.out.println("Ist das Wort EIMER in der Zeile vorhanden: " + m.findeWortZE("EIMER"));
System.out.println("Ist das Wort SSE in der Spalte vorhanden: " + m.findeWortSP("SSE"));
System.out.println("Ist das Wort SSE in der Zeile vorhanden: " + m.findeWortZE("SSE"));
System.out.println("Ist das Wort MENSA in der Spalte vorhanden: " + m.findeWortSP("MENSA"));
System.out.println("Ist das Wort MENSA in der Zeile vorhanden : " + m.findeWortZE("MENSA"));
System.out.println("Ist das Wort QUAL in der Spalte vorhanden: " + m.findeWortSP("QUAL"));
System.out.println("Ist das Wort QUAL in der Zeile vorhanden: " + m.findeWortZE("QUAL"));
System.out.println("Ist das Wort ARRAY in der Spalte vorhanden: " + m.findeWortSP("ARRAY"));
System.out.println("Ist das Wort ARRAY in der Zeile vorhanden: " + m.findeWortZE("ARRAY"));
System.out.println("Ist das Wort BRATEN in der Spalte vorhanden: " + m.findeWortSP("BRATEN"));
System.out.println("Ist das Wort BRATEN in der Zeile vorhanden: " + m.findeWortZE("BRATEN"));
}

public void ausgabe() {
    for (int x=0; x<matrix.length; x++) {
        for (int y=0; y<matrix.length; y++)
            System.out.print(matrix[x][y] + " ");
        System.out.println();
    }
}

public boolean findeWortSP(String wort) {
    boolean korrekt;
    for (int y=0; y<matrix[0].length; y++) {
        for (int x=0; x<matrix.length-wort.length(); x++) {
            if (matrix[x][y] == wort.charAt(0)) {
                korrekt = true;
                for (int i=0; i<wort.length(); i++)
                    if (matrix[x+i][y] != wort.charAt(i))
                        korrekt = false;
                if (korrekt == true)
                    return true;
            }
        }
    }
    return false;
}

public boolean findeWortZE(String wort) {
    boolean korrekt;
    for (int x=0; x<matrix.length; x++) {
        for (int y=0; y<matrix[x].length-wort.length(); y++) {
            if (matrix[x][y] == wort.charAt(0)) {
                korrekt = true;
                for (int i=0; i<wort.length(); i++)
                    if (matrix[x][y+i] != wort.charAt(i))
                        korrekt = false;
                if (korrekt == true)
                    return true;
            }
        }
    }
    return false;
}
}
```