

# Java programmieren: Musterlösungen Konsolen-Aufgaben

## Aufgabe 1: Gefüllte Rechtecke zeichnen

Schreiben Sie ein Programm, das ein durch Sternchen gefülltes Rechteck zeichnet. Der Benutzer soll Breite und Höhe eingeben können:

```
Geben Sie die Breite des Rechtecks ein: 20
Geben Sie die Höhe des Rechtecks ein:  5
*****
*****
*****
*****
*****
```

```
public class SternchenRechteckGefuellt {

    public static void main(String[] args) throws IOException {
        final BufferedReader konsolenEingabe = new BufferedReader(
            new InputStreamReader(System.in));
        System.out.print("Geben Sie die Breite des Rechtecks ein: ");
        final int breite= Integer.parseInt(konsolenEingabe.readLine());
        System.out.print("Geben Sie die Höhe des Rechtecks ein:  ");
        final int hoehe = Integer.parseInt(konsolenEingabe.readLine());

        for (int y = 0; y < hoehe; y++) {
            wiederholeZeichen(breite, '*');
            System.out.println();
        }

        static void wiederholeZeichen(final int breite, char c) {
            for (int x = 0; x < breite; x++) {
                System.out.print(c);
            }
        }
    }
}
```

## Aufgabe 2: Rechteck-Umrandung zeichnen

Schreiben Sie ein Programm, das ein durch Sternchen umrandetes Rechteck zeichnet. Der Benutzer soll Breite und Höhe eingeben können:

Geben Sie die Breite des Rechtecks ein: 20

Geben Sie die Höhe des Rechtecks ein: 4

```
*****
*                                     *
*                                     *
*****
```

```
public class SternchenRechteckUmrandung {

    public static void main(String[] args) throws IOException {
        final BufferedReader konsolenEingabe = new BufferedReader(
            new InputStreamReader(System.in));
        System.out.print("Geben Sie die Breite des Rechtecks ein: ");
        final int breite= Integer.parseInt(konsolenEingabe.readLine());
        System.out.print("Geben Sie die Höhe des Rechtecks ein: ");
        final int hoehe = Integer.parseInt(konsolenEingabe.readLine());

        if ((breite >= 2) && (hoehe >= 2)) {
            wiederholeZeichen(breite, '-');
            System.out.println();

            for (int y = 0; y < hoehe - 2; y++) {
                System.out.print('|');
                wiederholeZeichen(breite - 2, ' ');
                System.out.println('|');
            }

            wiederholeZeichen(breite, '-');
            System.out.println();
        }

        static void wiederholeZeichen(final int breite, char c) {
            for (int x = 0; x < breite; x++) {
                System.out.print(c);
            }
        }
    }
}
```

### Aufgabe 3: Unterschiede zwischen zwei Texteingaben zählen

Ein häufiges Problem ist der Vergleich von zwei Dateien: Wo unterscheiden Sie sich? Betrachten wir hier die Aufgabe, die Anzahl Unterschiede zwischen zwei eingegebenen Texten zu zählen.

```
public class AnzahlUnterschiede {

    public static void main(String[] args) throws IOException {
        final BufferedReader konsolenEingabe = new BufferedReader(
            new InputStreamReader(System.in));
        final String eingabe1 = konsolenEingabe.readLine();
        final String eingabe2 = konsolenEingabe.readLine();

        int anzahlUnterschiede = 0;

        int maxLaenge = Math.max(eingabe1.length(), eingabe2.length());
        for (int i = 0; i < maxLaenge; i++) {
            if (i < eingabe1.length() && i < eingabe2.length()) {
                if (eingabe1.charAt(i) != eingabe2.charAt(i)) {
                    anzahlUnterschiede++;
                }
            } else {
                anzahlUnterschiede++;
            }
        }

        System.out.println("Anzahl Unterschiede: " +
            anzahlUnterschiede);
    }
}
```

### Aufgabe 4: Berechnungen auf eingegebenen Zahlen

Schreiben Sie ein Programm, das eine Liste von Zahlen – getrennt durch einzelne Leerzeichen – entgegennimmt und die Summe sowie den Mittelwert der Zahlen ausgibt.

```
public class ZahlenAddieren {

    public static void main(String[] args) throws IOException {
        final BufferedReader konsolenEingabe = new BufferedReader(
            new InputStreamReader(System.in));
        final String eingabe = konsolenEingabe.readLine();
        final String[] zahlen = eingabe.split(" ");

        if (zahlen.length > 0) {
            int summe = 0;
            for (int i = 0; i < zahlen.length; i++) {
                summe = summe + Integer.parseInt(zahlen[i]);
            }
            System.out.println("Summe: " + summe);
            System.out.println("Mittelwert: " +
                (summe / zahlen.length));
        } else {
            System.out.println("Leere Eingabe, Abbruch.");
        }
    }
}
```

## Aufgabe 5: Doppellautanalyse

Schreiben Sie ein Programm, das die Anzahl Diphthonge (siehe auch <http://de.wikipedia.org/wiki/Diphthong>) in der Eingabe zählt. Wir betrachten hier die Diphthonge au, ei, ai.

```
public class DiphthongeZaehlen {  
  
    public static void main(String[] args) throws IOException {  
        final BufferedReader konsolenEingabe = new BufferedReader(  
            new InputStreamReader(System.in));  
        final String eingabe = konsolenEingabe.readLine();  
  
        int anzahlDiphtonge = 0;  
        for (int i = 0; i < eingabe.length() - 1; i++) {  
            char c1 = eingabe.charAt(i);  
            char c2 = eingabe.charAt(i + 1);  
            if (istDiphthong(c1, c2)) {  
                anzahlDiphtonge++;  
            }  
        }  
        System.out.println("Anzahl Diphthonge: " + anzahlDiphtonge);  
    }  
  
    static boolean istDiphthong(char c1, char c2) {  
        return ((c1 == 'a') && (c2 == 'u')) ||  
            ((c1 == 'e') && (c2 == 'i')) ||  
            ((c1 == 'a') && (c2 == 'i'));  
    }  
}
```

## Aufgabe 6: Vokale filtern

Schreiben Sie ein Programm, das alle Vokale aus der Eingabe filtert, also nur Nicht-Vokale wieder ausgibt; die Anzahl gefilterter Vokale soll ebenfalls ausgegeben werden:

```
public class VokaleWeglassen {  
  
    public static void main(String[] args) throws IOException {  
        final BufferedReader konsolenEingabe = new BufferedReader(  
            new InputStreamReader(System.in));  
        final String eingabe = konsolenEingabe.readLine();  
  
        int anzahlVokale = 0;  
        for (int i = 0; i < eingabe.length(); i++) {  
            char c = eingabe.charAt(i);  
            if (istKeinVokal(c)) {  
                System.out.print(eingabe.charAt(i));  
            } else {  
                anzahlVokale++;  
            }  
        }  
        System.out.println();  
  
        System.out.println("Anzahl weggelassener Vokale: " +  
            anzahlVokale);  
    }  
  
    static boolean istKeinVokal(char c) {  
        return (c != 'a') && (c != 'e') && (c != 'i') && (c != 'j')  
            && (c != 'o') && (c != 'u');  
    }  
}
```

## Aufgabe 7: Leerzeichen komprimieren

Schreiben Sie ein Programm, dass mehrere aufeinanderfolgende Leerzeichen durch ein einzelnes Leerzeichen ersetzt:

```
public class LeerzeichenReduzieren {  
  
    public static void main(String[] args) throws IOException {  
        final BufferedReader konsolenEingabe = new BufferedReader(  
            new InputStreamReader(System.in));  
        final String eingabe = konsolenEingabe.readLine();  
  
        boolean warVorherEinLeerzeichen = false;  
        for (int i = 0; i < eingabe.length(); i++) {  
            if (eingabe.charAt(i) == ' ') {  
                if (!warVorherEinLeerzeichen) {  
                    System.out.print(eingabe.charAt(i));  
                    warVorherEinLeerzeichen = true;  
                }  
            } else {  
                System.out.print(eingabe.charAt(i));  
                warVorherEinLeerzeichen = false;  
            }  
        }  
    }  
}
```