

# Zusatzaufgaben zum PGdP

Ich habe mal nach ein paar zusätzlichen alten Aufgaben gesucht, die ihr euch nochmal anschauen könnt, falls ihr Zeit habt. Durch die Routine sollte sich langsam auch ein besseres Verständnis für das Programmieren ergeben.

Bitte versucht die Aufgaben erstmal ohne euch den Kontrollfluss oder die Lösung anzusehen!

## 1 - Selektion

### 1.1: Wochentage

Erstelle ein Programm, dass folgende Anforderungen erfüllt:

1. Der Anwender gibt eine Ganzzahl zwischen 1 und 7 ein.
  2. Das Programm gibt den zugehörigen Wochentag aus.
- Prüfe die Eingabe auf Plausibilität und gebe falls nötig eine Fehlermeldung aus. Das Programm wird durch die Eingabe von 0 beendet.

```
<Bitte Zahl eingeben:
>5
<Freitag
<Bitte Zahl eingeben:
>9
<Fehler
<Bitte Zahl eingeben:
>-3
<Fehler
<Bitte Zahl eingeben:
```

### 1.2: Schaltjahre

Erstelle ein Programm, dass folgende Anforderungen erfüllt:

1. Der Benutzer gibt eine Jahreszahl zwischen 0 und  $\infty$  ein.
2. Das Programm gibt aus, ob es sich beim eingegebenen Jahr um ein Schaltjahr handelt.

Prüfe die Eingabe auf Plausibilität und gebe falls nötig eine Fehlermeldung aus.

Regeln für Schaltjahre (hierarchisch steigend):

- > Das Jahr lässt sich restlos durch 4 teilen → Schaltjahr
- > Das Jahr lässt sich restlos durch 100 teilen → kein Schaltjahr
- > Das Jahr lässt sich restlos durch 400 teilen → Schaltjahr

Das Programm wird durch die Eingabe von -5 beendet.

```
<Bitte Zahl eingeben:
>2000
<Schaltjahr
<Bitte Zahl eingeben:
>2001
<kein Schaltjahr
<Bitte Zahl eingeben:
>-3
<Fehler
<Bitte Zahl eingeben:
>1900
<kein Schaltjahr
<Bitte Zahl eingeben:
```

## 2 - Iteration

### 2.1: Zählen

Erstelle ein Programm, dass die Zahlen von 1 bis  $n$  ausgibt. Frage den Benutzer nach dem  $n \geq 1$ .

Prüfe die Eingabe auf Plausibilität und gebe falls nötig eine Fehlermeldung aus. *Das Programm läuft nur einmal durch.*

```
<Bitte Zahl eingeben:
>5
<1 2 3 4 5
<Bitte Zahl eingeben:
>0
<Fehler
<Bitte Zahl eingeben:
```

### 2.2: Gerade Zahlen

Erstelle ein Programm, dass die gerade Zahlen von  $n$  bis 1 (absteigend) ausgibt. Frage den Benutzer nach dem  $n \geq 1$ .

Prüfe die Eingabe auf Plausibilität und gebe falls nötig eine Fehlermeldung aus. *Das Programm läuft nur einmal durch.*

```
<Bitte Zahl eingeben:
>9
<8 6 4 2
<Bitte Zahl eingeben:
>-5
<Fehler
<Bitte Zahl eingeben:
```

**2.3: Collatz Zahlen**

Gebe die Zahlen der Collatz-Folge in deinem Programm aus. Starte mit einem  $n > 0$ , das der Benutzer eingibt. Die Collatz-Folge wird nach folgendem Bildungsgesetz gebildet:

- > Beginne mit einem  $n \in \mathbb{N} : n > 1$
- > Ist  $n$  gerade, so nimm als Nächstes  $n/2$
- > Ist  $n$  ungerade, so nimm als nächstes  $3n + 1$
- > Wiederhole die Vorgehensweise mit der erhaltenen Zahl
- > Wiederhole, bis die erhaltene Zahl 1 ist.

Prüfe die Eingabe auf Plausibilität und gebe falls nötig eine Fehlermeldung aus. *Das Programm läuft nur einmal durch.*

Das Programm sollte immer terminieren - sonst wäre das eine mathematische Sensation, denn warum die Collatz-Folge immer bei 1 terminiert, weiß man bis heute nicht (aber ihr hättet einen Gegenbeweis gefunden)... ↗[Wikipedia](#)

```
<Bitte Zahl eingeben:
>5
<5 16 8 4 2 1
<Bitte Zahl eingeben:
>-3
<Fehler
<Bitte Zahl eingeben:
>9
<9 28 14 7 22 11 34 17 52
 26 13 40 20 10 5 16 8 4
 2 1
<Bitte Zahl eingeben:
```

**3 - Arrays****3.1: Quadratzahlen**

Einem Array mit  $n$  Elementen (vom Benutzer erfragen) werden die jeweiligen Quadratzahlen von 0 bis  $\text{length} - 1$  (dem Index entsprechend) in einer Schleife hinzugefügt. Im Anschluss werde die Werte in einer zweiten Schleife ausgegeben.

Prüfe die Eingabe auf Plausibilität ( $n > 0$ ) und gebe falls nötig eine Fehlermeldung aus. *Das Programm läuft nur einmal durch.*

```
<Bitte Zahl eingeben:
>5
<0 1 4 9 16
<Bitte Zahl eingeben:
>-3
<Fehler
<Bitte Zahl eingeben:
>1
<0
<Bitte Zahl eingeben:
```

**3.2: Arithmetisches Mittel**

Erstelle ein Programm, das 3 Zahlen einließt und das arithmetische Mittel der Werte zurückgibt. *Das Programm läuft nur einmal durch.*

$$\text{arith. Mittel} = \frac{\sum_{i=1}^n a_i}{n}$$

```
<Bitte Zahl eingeben:
>1
<Bitte Zahl eingeben:
>7
<Bitte Zahl eingeben:
>10
<6
```

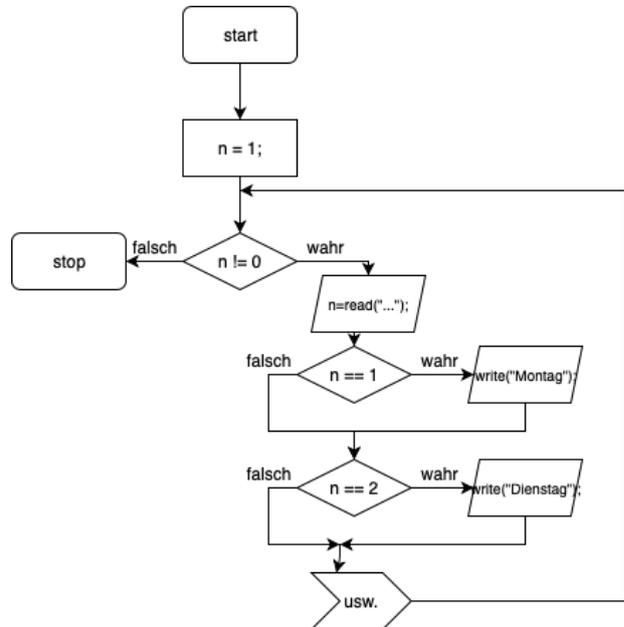
# Lösungen

Beachte: Falls in der Aufgabe steht: „Das Programm läuft nur einmal durch“, starten die zusätzlichen Durchläufe im Beispiel nicht automatisch sondern dienen nur dem besseren Verständnis. Die „**„**“ sind manchmal falsch - die sollen natürlich alle oben sein!

## 1 - Selektion

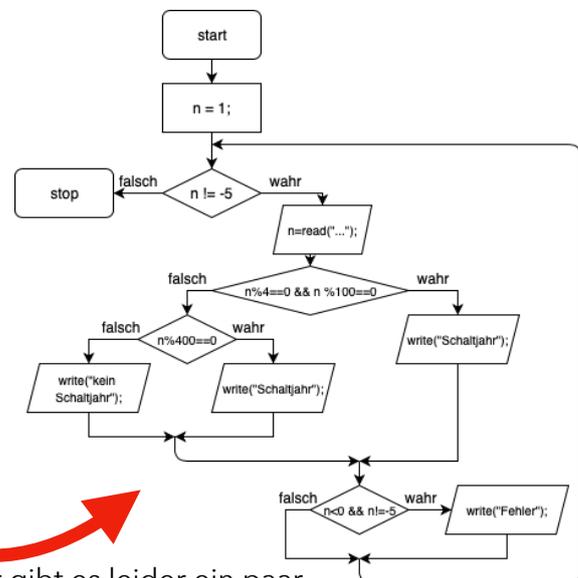
### 1.1: Wochentage

```
public static void main(String args[]) {
    int n = 1; //so läuft das Programm
    //erstmal in die Schleife rein
    while(n != 0) {
        n = readInt(„Bitte Zahl eingeben:“);
        if(n == 1) write(„Montag“);
        if(n == 2) write(„Dienstag“);
        if(n == 3) write(„Mittwoch“);
        if(n == 4) write(„Donnerstag“);
        if(n == 5) write(„Freitag“);
        if(n == 6) write(„Samstag“);
        if(n == 7) write(„Sonntag“);
        if(n < 0 || n > 7) write(„Fehler“);
    }
}
```



### 1.2: Schaltjahre

```
public static void main(String args[]) {
    int n = 0; //so läuft das Programm
    //erstmal in die Schleife rein
    while(n != -5) {
        n = readInt(„Bitte Zahl eingeben:“);
        if(n < 0 && n != -5) write(„Fehler“);
        else {
            if(n%4 == 0 && n%100 != 0)
                write(„Schaltjahr“);
            else {
                if(n%400 == 0)
                    write(„Schaltjahr“);
                else
                    write(„kein Schaltjahr“);
            }
        }
    }
}
```



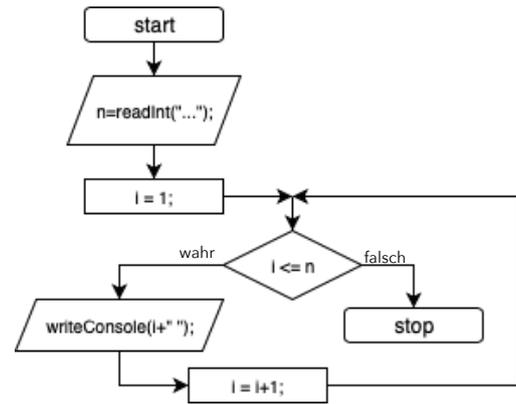
Hier gibt es leider ein paar Inkonsistenzen...

## 2 - Iteration

### 2.1: Zählen

```
public static void main(String args[]) {
    int n = readInt("Bitte Zahl eingeben:");
    int i = 1;
    while(i <= n) {
        writeConsole(i+" ");
        i = i + 1;
    }
}
```

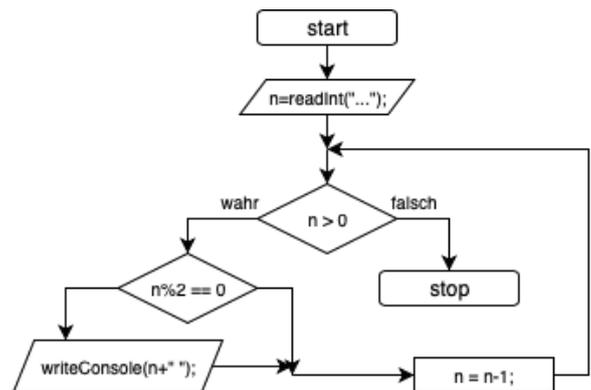
Plausibilitätsabfrage fehlt



### 2.2: Gerade Zahlen

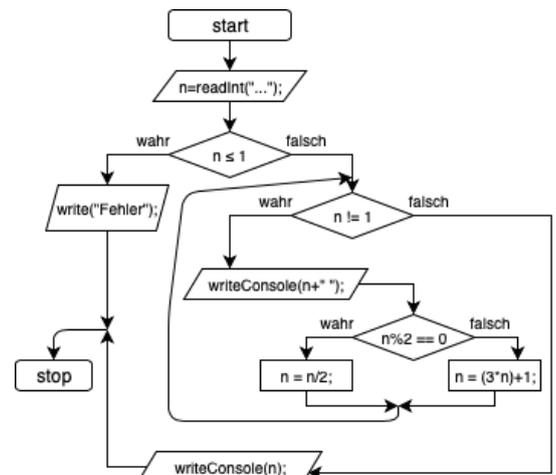
```
public static void main(String args[]) {
    int n = readInt("Bitte Zahl eingeben:");
    while(n > 0) {
        if(n%2 == 0)
            writeConsole(n+" ");
        n = n - 1;
    }
}
```

Plausibilitätsabfrage fehlt



### 2.3: Collatz Zahlen

```
public static void main(String args[]) {
    int n = readInt("Bitte Zahl eingeben:");
    if(n <= 1) write("Fehler");
    else {
        while(n != 1) {
            writeConsole(n+" ");
            if(n%2 == 0)
                n = n/2;
            else
                n = (3*n)+1;
        }
        writeConsole(n);
    }
}
```



## 3 - Arrays

### 3.1: Quadratzahlen

```
public static void main(String args[]) {
    int n = readInt("Bitte Zahl eingeben:");
    int[] a = new int[n--];
    while(n >= 0)
        a[n--] = (n+1)*(n+1); //Kurzschreibw.
    while(n < a.length-1)
        writeConsole(a[++n]+" ");
}
```

Plausibilitätsabfrage fehlt

### 3.2: Arithmetisches Mittel

```
public static void main(String args[]) {
    int[] a = new int[3];
    for(int i = 0; i < 3; i++) //for-Schleife
        a[i] = readInt();
    int sum = 0;
    for(int i = 0; i < 3; i++)
        sum += a[i];
    write(sum/3);
}
```