

Legende:

🌱 = einfach

🌱🌱 = mittel

🌶️🌶️🌶️ = schwer

🌱 Aufgabe 1

Schreibe ein Programm, das mithilfe einer for-Schleife die ersten zehn geraden, natürlichen Zahlen am Bildschirm ausgibt, also die Zahlen 2, 4, 6, 8, 10, 12, 14, 16, 18 und 20.

🌱 Aufgabe 2

Schreibe dein Programm aus Aufgabe 1 so um, dass es statt mit einer for-Schleife mit einer while-Schleife arbeitet.

🌱 Aufgabe 3

Schreibe dein Programm aus Aufgabe 1 so um, dass es statt mit einer for-Schleife mit einer do-while-Schleife arbeitet.

Hinweis für die nächsten Aufgaben:

Mithilfe der Methode `System.out.print()` kann man eine Ausgabe auf dem Bildschirm erzeugen, ohne dabei am Ende der Ausgabe einen Zeilenumbruch zu erhalten.

Beispiel:

```
System.out.print("Hello, ");
System.out.print("world");
System.out.println("!");
```

Ausgabe:

```
Hello, world!
```

Auf diese Art können mehrere Ausgaben in derselben Zeile erfolgen. Beachte auch, dass die `println()`-Methode **ohne Parameter** aufgerufen werden kann, um lediglich einen Zeilenumbruch zu erzeugen.

🌱🌱 Aufgabe 4

Schreibe eine Klasse `ChristmasTree`, die einen „Weihnachtsbaum“ darstellen soll. Dem Konstruktor der Klasse soll die Höhe des Baums als Ganzzahl übergeben werden (z. B. `ChristmasTree tree = new ChristmasTree (5);`). Die Klasse soll über eine Methode `print()` verfügen, die den Baum auf den Bildschirm „malt“. Für eine Höhe von fünf soll die Ausgabe wie folgt aussehen:

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

|

Aufgabe 5

Schreibe ein Programm, das eine positive Ganzzahl in seiner Binärdarstellung als 8-Bit-Zahl am Bildschirm ausgibt. Die Dezimalzahl 202 soll z. B. als 11001010 am Bildschirm ausgegeben werden. Du darfst davon ausgehen, dass die gegebene Zahl nicht größer als 255 ist (damit 8 Bit ausreichen, um die Zahl darzustellen).

Hinweis für die nächste Aufgabe:

Für diese Aufgabe wollen wir vom Benutzer oder der Benutzerin Eingaben von der Tastatur einlesen. Dafür können wir die Klasse `Scanner` benutzen, die Teil der Java-Standardbibliothek ist. Die Dokumentation dieser Klasse findet ihr unter:

<https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

Dem Konstruktor der Klasse `Scanner` müssen wir mitteilen, von wo die Eingaben gelesen werden sollen. Wenn die Eingaben von der Tastatur eingelesen werden sollen, geben wir dort `System.in` an (ebenfalls Teil der Java-Standardbibliothek).

Legt man ein Objekt vom Typ `Scanner` an, kann man mit dessen Methode `nextInt()` eine Zahl von der Tastatur einlesen. Beispiel:

```
// Scanner-Objekt erzeugen (Quelle ist die Tastatur)
Scanner scanner = new Scanner(System.in);

// Zahl von der Tastatur einlesen
int input = scanner.nextInt();
```

Wenn das `Scanner`-Objekt nicht mehr gebraucht wird, muss es „geschlossen“ werden. Das geschieht mit einem Aufruf der `close()`-Methode:

```
scanner.close();
```

Experimentiere bitte zunächst mit der `Scanner`-Klasse und schreibe eigene, einfache Programme, die Zahlen von der Tastatur einlesen.

Rufe dir auch noch einmal in Erinnerung, wie man in Java Zufallszahlen erzeugen kann (siehe Folien der letzten Vorlesung).

Aufgabe 6

Schreibe ein „Zahlen raten“-Spiel. Das Programm soll dabei eine Zufallszahl im Intervall $I = [1; 100]$ generieren und der Benutzer bzw. die Benutzerin soll die Zahl erraten. Dazu liest das Programm nacheinander Zahlen von der Tastatur ein und „bewertet“ die Eingabe („Die eingegebene Zahl ist zu groß“, „Die eingegebene Zahl ist zu klein“ oder „Du hast die richtige Zahl erraten“). Das Programm soll enden, sobald die korrekte Zahl eingegeben wurde.

Ein Programmdurchlauf könnte beispielsweise so aussehen:

```
Ich denke mir eine Zahl von 1 bis 100 aus. Du sollst sie erraten.
Bitte gib deinen Tipp ab:
```

43
Die eingegebene Zahl ist zu klein.
72
Die eingegebene Zahl ist zu groß.
61
Du hast die richtige Zahl erraten.

Aufgabe 7

Erweitere dein Programm aus Aufgabe 6: Die Anzahl der Versuche soll nun auf sechs begrenzt werden. Das bedeutet, dass man das Spiel verliert, wenn man mehr als sechs Versuche benötigt.

Aufgabe 8

Schreibe ein Programm, das alle Primzahlen im Intervall $I = [1; 100]$ am Bildschirm ausgibt.