

Programmieraufgaben

1. Hello World! Schreibe ein Programm, das auf der Konsole den Text „Hello World!“ ausgibt.	Textausgabe auf Bildschirm ¹
2. Netto-Brutto-Rechner: Schreibe ein Programm, bei dem der Nutzer einen Preis ohne Steuer eingibt. Der Rechner gibt den Preis mit 19% Mehrwertsteuer aus.	Variablendeklaration ² , Eingabe über Tastatur ³ , Textausgabe auf Bildschirm
3. Zinsberechnung: Der Nutzer soll einen Geldbetrag und einen Zinssatz eingeben können. Ausgabe sollen die Zinsen für ein Jahr sein.	
4. Rabattrechner: Eingabe soll der Normalpreis sein und wie viel Prozent Rabatt es auf den Artikel gibt. Ausgabe soll der Endpreis sein.	
5. Begrüßung: Schreibe ein Programm, das den Nutzer nach seinem Namen fragt und ihn dann mit seinem Namen begrüßt, z.B. „Hallo Johanna! Wie geht's dir?“	Variablendeklaration ⁴ , Eingabe über Tastatur ⁵ , Textausgabe auf Bildschirm
6. Quader: Schreibe ein Programm, bei dem der Nutzer Breite, Länge und Höhe eines Quaders angibt. Das	Variablendeklaration, Tastatureingabe,

- 1 **Ausgabe auf dem Bildschirm:** `System.out.println(„Hello World“)` ; Die Buchstaben, die innerhalb der Anführungszeichen stehen, werden genau so auf dem Bildschirm ausgegeben. `System.out.println(v)` ; Steht in der Klammer etwas ohne Anführungszeichen, so muss es sich hierbei um einen Variablennamen handeln. Es wird dann der **Inhalt** der Variablen ausgegeben, nicht ihr Name.
- 2 **Variablendeklaration:** Verwendet man eine Variable, also einen Speicherplatz für eine Zahl oder ein Wort, so muss man dem Computer sagen, was für eine Variable man benötigt. Wie soll sie heißen und was ist ihr Typ? Z.B. `String name`; für eine Zeichenkette, also mehrere Buchstaben hintereinander oder `int i`; für eine Ganzzahl oder `double x`; für eine Kommazahl.
- 3 **Eingabe über Tastatur:** Soll der Nutzer etwas über die Tastatur eingeben, muss eine Variable deklariert worden sein, in der der Inhalt gespeichert werden soll. Z.B. eine Double-Variable `double laenge`; Die Eingabe über die Tastatur erfolgt dann mit der Zuweisung `laenge= InOut.readDouble(„Geben Sie die Länge ein:“)` ; Weiteres Beispiel für einen String, der eingegeben wird: `String name; name=InOut.readString(„Geben Sie Ihren Namen ein:“)` ; Falls du nicht mit dem Javaeditor arbeitest, muss die Datei `InOut.java` im gleichen Verzeichnis liegen. Diese kannst du auf Moodle herunterladen.
- 4 **Variablendeklaration:** Verwendet man eine Variable, also einen Speicherplatz für eine Zahl oder ein Wort, so muss man dem Computer sagen, was für eine Variable man benötigt. Wie soll sie heißen und was ist ihr Typ? Z.B. `String name`; für eine Zeichenkette, also mehrere Buchstaben hintereinander oder `int i`; für eine Ganzzahl oder `double x`; für eine Kommazahl.

Programm berechnet dann das Volumen, die Oberfläche und die Raumdiagonale ($\sqrt{a^2+b^2+c^2}$)	Rechnungen ⁶ , Wurzel ⁷)
7. Quaderberechnung abändern: Ändere das Programm Quaderberechnung so ab, dass es den Nutzer darauf hinweist, falls er negative Eingaben macht, dass dies nicht sein kann.	
8. Rabattrechner abändern: Weise den Nutzer darauf hin, wenn er 100 oder mehr Prozent eingibt, dass dies nicht sein kann.	
9. Rückgeld: Schreibe ein Programm, bei dem man eingibt, wie viel jemand zahlen muss und wie viel er in bar gezahlt hat. Es soll ausgegeben werden, wie viel Rückgeld der Kunde bekommt. Falls er zu wenig bezahlt hat, soll das Programm eine Warnung ausgeben.	
10. Das Programm soll eine quadratische Gleichung der Form $ax^2+bx+c=0$ lösen können. Der Nutzer gibt hierfür die Zahlen a, b, c ein. Das Programm prüft die Diskriminante, also b^2-4ac , ob diese positiv, null oder negativ ist und gibt dann aus wie viele Lösungen es gibt und ggf. deren Werte.	Tastatureingabe, Variablendeklaration, Rechnungen, Wurzel, Verzweigung ⁸
11. BMI-Rechner: Eingabe: Körpergröße, Körpergewicht und Geschlecht. Ausgabe: BMI und Hinweis, ob Unter-, Normal- oder Übergewicht. (Bei Frauen ist ein BMI von 19 bis 24 normal, bei Männern 20 bis 25.)	Deklaration, Tastatureingabe, Rechnung, Verzweigung

- 5 **Eingabe über Tastatur:** Soll der Nutzer etwas über die Tastatur eingeben, muss eine Variable deklariert worden sein, in der der Inhalt gespeichert werden soll. Z.B. eine Double-Variable `double laenge`; Die Eingabe über die Tastatur erfolgt dann mit der Zuweisung `laenge= InOut.readDouble(„Geben Sie die Länge ein:“)`; Weiteres Beispiel für einen String, der eingegeben wird: `String name; name=InOut.readString(„Geben Sie Ihren Namen ein:“)`; Falls du nicht mit dem Javaeditor arbeitest, muss die Datei `InOut.java` im gleichen Verzeichnis liegen. Diese kannst du auf Moodle herunterladen.
- 6 **Berechnungen:** Will man die Inhalte der Variablen `a`, `b`, `c` miteinander multiplizieren und das Ergebnis in der Variablen namens `volumen` speichern, ist der Befehl hierfür `volumen= a* b*c`;
- 7 **Quadratwurzel:** Um die Quadratwurzel berechnen zu können, muss die Mathematik-Bibliothek importiert werden: `import java.math.*`; Die Wurzel aus $a^2+b^2+c^2$ wird dann wie folgt berechnet: `d=Math.sqrt(a*a+b*b+c*c)` ;
- 8 **Verzweigung:** Soll das Programm prüfen, ob eine Bedingung erfüllt ist, z.B. ob der Wert der Variablen namens `diskriminte` negativ ist, dann ist die Java-Syntax folgende: `if (Bedingung) {Anweisungen}` also z.B. `if (diskriminante < 0) { System.out.println(„Es gibt keine Lösung.“) ;}`

12. Mittelwert: Schreibe ein Programm, bei dem man 4 Zahlen eingeben kann. Das Programm berechnet den Mittelwert.	
13. Wahlprogramm: Schreibe ein Programm, bei dem man für drei Kandidaten Namen eingeben kann und die Anzahl der erhaltenen Stimmen. Das Programm gibt die prozentualen Anteile der Stimmen aus.	
14. Maklerprogramm: Eingabe Grundstückslänge und -Breite, Quadratmeterpreis, Maklerprovision in %, aktueller Umsatzsteuersatz in %. Ausgabe: Grundstückspreis und Maklerprovision.	
15. Benzin-Rechner: Eingabe: Literpreis Benzin, verbrauchte Liter, gefahrene Kilometer. Ausgabe: Verbrauch pro 100 km, Preis pro 100 km, Preis der Gesamtstrecke.	Deklaration, Tastatureingabe, Rechnung
16. Kopfrechnen: Schreibe ein Programm, das zwei Zufallszahlen erzeugt (<code>a= (int) (Math.random() * 50) ;</code>) und den Nutzer auffordert diese zu addieren. Falls der Nutzer richtig gerechnet hat, soll dieser gelobt werden.	
17. Potenzrechnung: Schreibe ein Programm, bei dem der Nutzer zwei Zahlen a und b eingeben kann, b ganzzahlig. Das Programm berechnet die Potenz a^b .	
18. Fakultät berechnen: In der Wahrscheinlichkeitsrechnung werdet ihr später das Produkt aller natürlichen Zahlen bis n benötigen „n!“, sprich „n Fakultät“. Z.B. $3! = 1 \cdot 2 \cdot 3 = 6$ und $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$. Schreibe ein Programm, bei dem man n eingibt und das n! ausgibt.	
19. Größte Zweierpotenz: Schreibe ein Programm, bei dem der Nutzer eine natürliche Zahl eingibt und das Programm gibt die größte Zweierpotenz zurück, die kleiner oder gleich der eingegebenen Zahl ist.	While-Schleife
20. Schreibe ein Programm, das mithilfe einer Schleife n Zeilen mit 10 Sternen schreibt: ***** ***** *****	For-Schleife
21. Schreibe ein Programm, das mithilfe von zwei Schleifen n Zeilen mit je m Sternen schreibt. (Nutzer gibt n und m ein.)	geschachtelte For-Schleife
22. Halber Weihnachtsbaum: Schreibe ein Programm das einen „Weihnachtsbaum“ der Höhe n ausgibt. Eingabe: n.	

<pre> * ** *** **** </pre>	
<p>23. Schreibe ein Programm, das einen ganzen Weihnachtsbaum ausgibt:</p> <pre> * *** ***** ***** # </pre>	
<p>24. BugTrouble: Wir gehen davon aus, dass ein sehr vermehrungsfreudiger Käfer ein Volumen von $0.0002m^3$ hat und sich pro Woche um 95% vermehrt. Nun ist ein Programm zu schreiben, in das der Nutzer die momentan vorhandene Anzahl Käfer sowie das Volumen des Hauses eingibt. Das Programm soll dann für Woche 1, Woche 2 usw. die Anzahl der vorhandenen Käfer sowie deren Gesamtvolumen angeben – für so viele Wochen, bis das Haus komplett gefüllt ist.</p>	Deklaration, Tastatureingabe, Rechnungen, While-Schleife ⁹
<p>25. Kopfrechnen II: Verändere das Kopfrechnenprogramm so, dass es nicht nur eine Additionsaufgabe stellt, sondern 10. Es soll mitzählen, wie viele Aufgaben der Nutzer richtig hatte.</p>	

9 **While-Schleife:** Soll ein Block von Anweisung mehrmals hintereinander ausgeführt werden, solange wie eine bestimmte Bedingung gilt, so ist die Java-Syntax folgende: **while (Bedingung) {Anweisungsfolge, die wiederholt werden soll}** also z.B. **while (kaeferVolumen < hausVolumen) { Berechnungen und Ausgabe }**

<p>26. Schreibe ein Programm, dass auf Wunsch des Nutzers eine Multiplikationsreihe ausgibt. Der Nutzer soll angeben können, welche Reihe, z.B. die ger-Reihe, also $1*9=9$; $2*9=18$; $3*9=27$ usw. Der Nutzer soll außerdem angeben können, wie weit die Reihe gehen soll.</p>	For-Schleife ¹⁰
<p>27. Fibonacci-Zahlen: Schreibe ein Programm, das die Fibonaccizahlen bis n (Eingabe) ausgibt: 0 1 1 2 3 5 8 13 21 34 55 89 144 usw. Die nächste Zahl ergibt sich immer als Summe der beiden vorangegangenen.</p>	While-Schleife
<p>28. Rückgeld: Schreibe ein Konsolenprogramm, bei dem der Nutzer einen Geldbetrag mit Euro und Cent eingibt und der Rechner ausgibt mit wie vielen 50-Euro-Scheinen, 20-Euro-Scheinen, 10-Euro-Scheinen usw. bis wie viele 1-Cent-Stücke,</p> <pre data-bbox="331 813 1114 1086"> Geben Sie die Anzahl Euros gefolgt von der Cent-Angabe ein: 89 30 Sie müssen 89 Euro 30 zurückgeben. Hundert-Euro-Scheine: 0. Bleiben 89 Euro. Fünfundzwanzig-Euro-Scheine: 1. Bleiben 39 Euro. Zwanzig-Euro-Scheine: 1. Bleiben 19 Euro. Zehn-Euro-Scheine: 1. Bleiben 9 Euro. Fünf-Euro-Scheine: 1 Zwei-Euro-Stücke: 2 Ein-Euro-Stücke: 0 </pre> <p>dieser Betrag ausbezahlt wird.</p>	Ganzzahlige Division ¹¹
<p>29. Zeitumrechnung: Eingabe: Anzahl Sekunden; Ausgabe: Anzahl Tage, Stunden, Minuten, Sekunden.</p> <pre data-bbox="319 1256 1209 1406"> Geben Sie die Anzahl der Sekunden ein: 216512 Das sind 2 Tage, 12 Stunden, 8 Minuten und 32 Sekunden. </pre>	Ganzzahlige Division

¹⁰ **For-Schleife:** Sollen die gleichen Anweisungen mehrmals wiederholt werden und steht vor der Schleifenausführung schon fest, wie oft, kann man hierfür eine For-Schleife verwenden: `for (i=1; i<=10; i=i+1) {Anweisungen}` Die Anweisungen, die im Schleifenkörper stehen, werden nun für $i=1$, dann $i=2$, dann $i=3$ usw. bis 10 ausgeführt. Wollte man nur alle geraden Zahlen nehmen, müsste man das dritte Argument in den runden Klammer in $i=i+2$ ändern. Wollte man, dass die Schleife rückwärts läuft von $i=10$ bis $i=1$, so würde man in die runden Klammer `(i=10; i>=1; i=i-1)` schreiben. Das erste Argument gibt also den Startwert der Zählvariable i , das zweite die Bedingung, wie lange die Schleife laufen soll, das dritte, wie gezählt werden soll, also wie aus dem alten i das neue i berechnet wird.

¹¹ **Ganzzahlige Division:** Sind die Variablen `euro` und `anzahl` beide als integer deklariert, führt der Befehl `anzahl = euro/50`; eine ganzzahlige Division durch, d.h. in `anzahl` wird gespeichert, wie oft die 50 ganz in `euro` hineinpasst. Z.B. `96/50` gibt 1. Den Rest erhält man mit `96%50`, also hier 46.

<p>30. Kopfrechnen (nur Addition): Das Programm stellt 10 Kopfrechenaufgaben. Das Programm erzeugt hierfür je zwei Zufallszahlen, die der Nutzer im Kopf addieren soll. Das Programm zählt, wie viele Ergebnisse der Nutzer richtig hat und zeigt dies an.</p>	<pre> Addition üben: Aufgabe 1: 26+10=36 Prima! Aufgabe 2: 22+30=52 Prima! Aufgabe 3: 30+18=48 Prima! Aufgabe 4: 20+9=29 Prima! Aufgabe 5: 17+26=43 Prima! Aufgabe 6: 17+13=30 Prima! Aufgabe 7: 1+21=22 Prima! Aufgabe 8: 5+3=8 Prima! Aufgabe 9: 11+11=22 Prima! Aufgabe 10: 14+6=1 Leider falsch! Richtig gewesen wäre: 20 9 von 10 Aufgaben richtig. </pre>	<p>Zufallszahlen</p>
<p>31. Kopfrechnen advanced: Erweitere das Programm so, dass der Nutzer zwischen +, -, *, : wählen kann.</p>		<p>For-Schleife, Verzweigung</p>

- 12 **Zufallszahlen erzeugen:** Wenn `java.math.*` importiert wurde, kann man mit `Math.random()` eine Zufallszahl vom Typ `double` erzeugen, die zwischen 0 und 1 ist. Multipliziert man diese Zufallszahl mit 30 hat man eine zufällige Kommazahl zwischen 0 und 30. Wandelt man diese mit `(int)` in einen Integer um, hat man als eine zufällige Ganzzahl zwischen 0 und 30. Insgesamt lautet dann der Befehl: `a= (int) (Math.random() *30) ;`