



Erstellen Sie für JEDES Programm, bevor Sie dieses am Computer erstellen, das Stuktogramm.

Halten Sie beim Programmieren rigoros ALLE aufgestellten Programmierregeln ein. Ihre Programme werden in Hinsicht auf die Einhaltung dieser Regeln kontrolliert.



1. Beschreiben Sie die Funktionsweise folgender *Methoden* der Klasse Math. Was machen die einzelnen Methoden?

Math.abs, Math.ceil, Math.floor, Math.random

2. Von welchem *Datentyp* sind die Ergebnisse der folgenden Ausdrücke, und wie lauten ihre *Ergebnisse*? Machen Sie sich zuerst über den Typ des Ergebnisses Gedanken, bevor Sie die Ergebnisse ermitteln:

	Typ	Ergebnis
3 + 2 * 7 + 2 / 3		
7 + 8 % 3		
6D + 3 % 8		
Math.pow(10 % 70, Math.sqrt(Math.sqrt(16)))		
1 + Math.sqrt(Math.pow(2, 4))		



3. Schreiben Sie ein *kurzes Testprogramm* mit dem Namen IntUeberlauf, welches Ihnen veranschaulicht was passiert, wenn in eine **int**-Variable ein Wert eingetragen wird, welcher den Wertebereich sprengt. Damit wird ein sogenannter *Überlauf* produziert. Schreiben Sie dazu zuerst in die Variable den *größtmöglichen Wert* hinein, und erhöhen Sie dann diesen um 1. Welcher Wert steht dann in der Variablen?

4. Führen Sie dasselbe Experiment auch für eine **double**-Variable durch. Das Programm das Sie dazu schreiben sollen, soll den Namen DoubleUeberlauf haben. Nehmen Sie hier wiederum den größtmöglichen **double** -Wert, und schreiben Sie diesen zuerst in die Variable. Dann erhöhen Sie diesen in einer *Schleife* – die beispielsweise 100000mal wiederholt wird – um einen vergleichsweise kleinen Faktor (z. B. um 1E100). Welcher Wert steht nach den wiederholt durchgeführten Erhöhungen in der Variable?

5. Sie sollen mit Hilfe eines Programms herausfinden, welches die *kleinste positive Doublezahl* ist. **Anleitung:** Fangen Sie mit der Zahl 1 an und halbiere sie so lange, bis sie Null ist. Die letzte von 0 verschiedene Zahl ist die gesuchte Zahl. Das zu erstellende Programm soll den Namen kleinstesDouble haben.

6. Addieren Sie mit *Doublezahlen* innerhalb einer Schleife 100 mal 0.1 auf. Schreiben Sie dazu ein Programm mit dem Namen HundertMalDouble. Welchen Wert hat die Summe? Warum hat die Summe nicht den Wert 10?

7. Betrachten Sie folgende Schleife:

```
double n = 10.0;
double i = 0.0;
while (i != n) {
    i = i + 0.1;
}
```

Anmerkung: i != n bedeutet i ungleich n

Wieso bricht diese Schleife nie ab? Schreiben Sie zur Prüfung eventuell ein *kurzes Testprogramm* das den Namen SchleifenAbbruch hat, und formulieren Sie Ihre Antwort.

Exkurs Benutzereingabe über Tastatur

Um dem Benutzers des Programms die Möglichkeit zu geben, Daten über *Tastatur einzugeben*, welche das Programm dann verarbeiten soll, wurde die einfach auf unsere Zwecke zugeschnittene Klasse `TestScanner` bereitgestellt, welche die Methoden `readInt` und `readDouble` enthält. Diese beiden Methoden erlauben die Eingabe eines Ganz- bzw. Gleitkommazahlenwertes.

```
TestScanner
=====
Geben Sie einen int-Wert ein: 3
Geben Sie einen double-Wert ein: 7,5
i = 3, d = 7.5
```

Betrachten Sie die Kommentare der Klasse, und verwenden Sie in den nachfolgenden Programmen diese beiden Methoden, um Benutzereingaben von Tastatur aus zu ermöglichen.

Hinweis: Beachten Sie das ein Gleitkommawert mit Komma (,) eingegeben werden muss, während dem die Ausgabe desselben Wertes mit Punkt (.) erfolgt.

8. Erstellen Sie das Programm mit dem Namen `Kugelberechnung`, welches *exakt* (!) die nebenstehende Benutzerschnittstelle hat, und das den *Umfang*, die *Oberfläche* und das *Volumen* der Kugel ermittelt. Der *Radius* wird über die Tastatur eingegeben.

```
Kugelberechnung
=====
Geben Sie den Radius der Kugel ein: 3,5

Der Umfang der Kugel beträgt: 21.991148575128552
Die Oberfläche der Kugel beträgt: 153.93804002589985
Das Volumen der Kugel beträgt: 179.5943800302165
```

Hinweis: Um die *Kreiszahl* π möglichst exakt in den Berechnungen verwenden zu können, stellt Java diese anhand von `Math.PI` zur Verfügung. Sie sollen diese in Ihren Berechnungen verwenden.

9. Schreiben Sie ein Programm mit dem Namen `Zinseszinsen`, welches für ein *Anfangskapital* (K_0), einen *Zinssatz* (p) und eine *Anzahl an Jahren* (n) das *Endkapital* (K_n) ermittelt, welche entsteht, wenn man das Anfangskapital über diesen Zeitraum mit *Zinseszinsen* verzinst. Diese drei Werte sollen über die Tastatur eingegeben werden.

```
Zinseszinsen
=====
Geben Sie das Anfangskapital ein: 10000
Geben Sie den Zinssatz ein: 3,75
Geben Sie die Jahre ein: 10

Das Endkapital nach 10 Jahren beträgt 14450
```

Dabei sollen die *Jahre* als *Ganzzahlwert* eingegeben werden und das *Endergebnis ohne Kommastellen* in *abgerundeter Form* ausgegeben werden. Das Programm soll wiederum exakt die nebenstehende Benutzerschnittstelle haben.

10. Sie sollen ein Programm mit dem Namen `Sparen in Raten` schreiben, welches das Guthaben eines Sparbuches nach einem Jahr berechnet, wenn jedes Monat dieselbe *Monatsrate* auf das Sparbuch einbezahlt wird.

```
Sparen in Raten
=====
Geben Sie das Guthaben am Jahresanfang ein: 1200
Geben Sie die Monatsrate ein: 50
Geben Sie den Jahreszinssatz ein: 3,125

Das Guthaben am Ende des Jahres beträgt 1847.65
```

Das Guthaben am Jahresende wird folgendermaßen berechnet:

$$\text{Guthaben am Ende des Jahres} = \text{Guthaben am Jahresanfang} + \text{Zinsen für das Guthaben am Jahresanfang} + 12 * \text{Raten} + \text{Zinsen für die 12 Raten}$$

Beachten Sie bei den *Raten*

- dass die 1. Rate sich mit $\frac{12}{12}$ der Jahreszinsen verzinst wird,
- dass die 2. Rate sich mit $\frac{11}{12}$ der Jahreszinsen verzinst wird,

dass die 3. Rate sich mit $\frac{10}{12}$ der Jahreszinsen verzinst wird,
 dass die 4. Rate sich mit $\frac{9}{12}$ der Jahreszinsen verzinst wird,
 ...
 dass die 11. Rate sich mit $\frac{2}{12}$ der Jahreszinsen verzinst wird,
 dass die 12. Rate sich mit $\frac{1}{12}$ der Jahreszinsen verzinst wird.

Das Guthaben am Jahresende soll *abgerundet auf zwei Kommastellen* ausgegeben werden.

11. Ein Satellit funkt Zeitspannen als *Anzahl Sekunden* zur *Satellitenzeit*
 Erde. Schreiben Sie ein Programm mit dem Namen *=====*
SatellitenZeit, das einen Sekundenbetrag einliest *Geben Sie die Sekunden ein: 123456*
 und die Zeitspanne in der Form d h m s wieder gibt. *Die umgerechnete Zeit ist:*
 Die Benutzerschnittstelle soll exakt wie abgebildet *d 1 h 10 m 17 s 36*
 dargestellt werden.

Dabei ist

d = Anzahl Tage,
 h = Anzahl Stunden im Bereich 0 bis 23,
 m = Anzahl Minuten im Bereich 0 bis 59,
 s = Anzahl Sekunden im Bereich von 0 bis 59.

12. Das Programm *Quersumme* soll die *Quersumme* einer *Quersumme*
 eingegebenen positiven Ganzzahl ermitteln. Die *=====*
 Benutzerschnittstelle soll wie angegeben gestaltet werden. *Geben Sie die Zahl ein: 12345*

Die Quersumme lautet:
1 + 2 + 3 + 4 + 5 = 15

13. Schreiben Sie ein weiteres Programm mit dem Namen *Quersummenquersumme*
QuersummenQuersumme, welches – falls möglich – aus *=====*
 der *Quersumme* die *Quersumme* ermittelt. Wiederum *Geben Sie die Zahl ein: 12345*
 soll das Programm exakt die angegebene *Die Quersummenquersumme lautet:*
 Benutzerschnittstelle haben und die Quersummen *1 + 2 + 3 + 4 + 5 = 15 = 1 + 5 = 6*
 anschaulich in der angegebenen Form ausgeben.

14. Schreiben Sie ein Programm mit dem Namen *Würfelsimulation*
wuerfelSimulation, welches 1.000.000.000-mal zufällig *=====*
 eine Zahl zwischen 1 und 6 ermittelt und dabei zählt, wie oft *Bitte warten*
 die einzelne Zahl gewürfelt wird. Das Programm soll die *.....*
 nebenstehende Benutzerschnittstelle haben.

Anzahl 1: 166652032
Anzahl 2: 166656486
Anzahl 3: 166674941
Anzahl 4: 166670933
Anzahl 5: 166660603
Anzahl 6: 166685005

Da das Durcharbeiten der Schleife – je nach
 Verarbeitungsgeschwindigkeit Ihres Computers – längere Zeit
 in Anspruch nehmen kann, soll ein einfacher
Fortschrittsbalken ausgegeben werden, damit der Benutzer
 erkennen kann, dass das Programm arbeitet. Der
 Fortschrittsbalken soll aus maximal 30 Punkten bestehen.

Anzahl Würfe: 1000000000

Eine Zufallszahl kann mit der Methode `Math.random` ermittelt
 werden. Beachten Sie, dass alle Zahlen mit der gleichen Wahrscheinlichkeit gewürfelt werden. Das
 bedeutet, dass für die Anzahl aller sechs Zahlen ungefähr dasselbe Ergebnis erzielt werden muss.

15. Das Programm weihnachtsbaum soll einen 15 Zeilen hohen Weihnachtsbaum mit einem 3 Zeilen hohen und 3 Sterne breiten Stamm am Bildschirm ausgeben.

Dazu sollen Sie zuerst folgende *Methoden programmieren* und dann bei der Ausgabe des Weihnachtsbaumes *verwenden*:

```
/**
 * Gibt anzahl Leerzeichen nebeneinander am
 * Bildschirm aus. Dabei wird nach dem
 * letzten ausgegebenen Leerzeichen keine
 * Zeilenschaltung gemacht
 * @param anzahl der auszugebenden
 * Leerzeichen
 */
public static void printLeerzeichen
(int anzahl) {
    ...
}

/**
 * wie vorige Methode nur mit Sternen
 */
public static void printSterne (int anzahl) {
    ...
}
```

15 Zeilen

3 Zeilen

16. Schreiben Sie ein Programm mit dem Namen `Der flexible Weihnachtsbaum`, beim welchem man die *Höhe* des Weihnachtsbaumes und die *Höhe des Stammes* eingeben kann. Das Programm soll dann den entsprechenden Baum zeichnen.

```

      古
    古 古 古
  古 古 古 古 古
    古 古 古
      古

```