

0. Aufgabe

Installieren Sie auf Ihrem Heimrechner das **Java-JDK** von Sun, welches von der Website der Firma Sun geladen werden kann. Installieren Sie auch die entsprechende **HTML-Dokumentation**. Diese Aufgabe soll Sie mit den grundlegenden Schritten bei der Programmentwicklung vertraut machen. Die **folgenden Schritte** sollen **nur im Kommandozeilen-Modus** (DOS-Shell) ausgeführt werden. Verwenden Sie die grafische Windows-Oberfläche höchstens zur Kontrolle.

- (a) Erstellen Sie zunächst eine für Sie sinnvolle Verzeichnis-Struktur! Sie können z.B. unter einem Verzeichnis **gdi** eine zweite Verzeichnis-Hierarchie einführen, welche die Kapitel darstellt und darunter eine dritte Verzeichnis-Hierarchie, welche die Aufgaben repräsentiert.
- (b) Rufen Sie den **DOS-Standard-Editor notepad** auf bzw. die ältere Variante **edit auf!** Machen Sie sich damit vertraut!
- (c) **Editieren** Sie damit das „Hello-World“-Beispiel aus der Vorlesung, welches in Kapitel 4.1, Seite 7 vorgestellt wurde. Denken Sie bitte daran, die richtige Datei-Endung zu verwenden. Nach dem Schließen des Editors können Sie das Ergebnis z.B. mit Hilfe des DOS-Befehls **more** überprüfen.
- (d) Rufen Sie den **Java-Compiler** auf! Welche Dateien sind neu entstanden? Verwenden Sie dazu den DOS-Befehl **dir!**
- (e) Führen Sie das gerade erstellte Java-Programm aus, d.h. **interpretieren** Sie also den entsprechenden **Java-Byte-Code!**
- (f) Lassen Sie sich auf der DOS-Shell den Inhalt der Umgebungsvariablen **Classpath** anzeigen! Bei der Verwendung benutzen Sie auch den expliziten Weg: Dazu bitte vorher die Umgebungsvariable Classpath auf Leer setzen und Classpath direkt beim Aufruf des Compilers bzw. Interpreters für ein Java-Programm angeben: **javac -classpath c:\java; .**

1. Aufgabe

Welche der folgenden Bezeichner sind ungültig?

Hallo_Welt \$Test _abc 2test #hallo te?st Girokonto const

2. Aufgabe

In Anlehnung an das „Hello World“- Beispiel aus Kapitel 3.1 schreiben Sie bitte ein „Hallo Uebung4“ Beispiel,

- (a) welches kompiliert und ausgeführt werden soll.
- (b) Verändern Sie die Deklaration der main-Methode. Was passiert in den folgenden Fällen, welche Fälle lassen sich compilieren, welche lassen sich ausführen?
- public static void main ()
 - public static void main (String args [])
 - public static void main (string [] args)
 - public static void main (String args)
 - public static void main (String [] feld)
 - public static void haupt (String [] args)
 - public static void Main (String [] args)
 - public static int main (String [] args)

3. Aufgabe

Warum führt der folgende Code zu einem Compiler-Fehler?

```
int x = 0;  
long y = 1000;  
x = y;
```

4. Aufgabe

Schreiben Sie ein Java-Programm, welches die Anzahl der Sekunden im Monat Januar berechnet.

5. Aufgabe

Schreiben Sie ein Programm, welches ermittelt, wie hoch ein Sparguthaben von 5000 EUR nach Ablauf eines Jahres ist bei einer 7,5% Verzinsung.

6. Aufgabe

Welche Werte haben die folgenden Ausdrücke und welche Werte haben die Variablen nach der Auswertung, wenn a den Anfangswert 1 und b den Anfangswert 7 hat?

- (a) --a
- (b) a--
- (c) a++ + b
- (d) b = ++a
- (e) a = b++
- (f) -(a--) - - (--b)

7. Aufgabe

Schreiben Sie ein Java-Programm, welches eine bestimmte Anzahl von Sternchen (*) in Form eines Dreiecks auf dem Bildschirm ausgibt. Geben Sie dabei eine feste Anzahl an Zeilen vor, z.B. n = 10.

8. Aufgabe

Schreiben Sie ein Java-Programm, welches den Buchwert in Höhe von 15.000 EUR mit einem Abschreibungssatz von 40% und einem Restwert von 100 EUR geometrisch degressiv abschreibt. Geben Sie dabei die entsprechenden Jahreswerte an, also den Buchwert nach dem 1. Jahr, 2. Jahr usw. bis der Restwert erreicht wird.

9. Aufgabe

Um die Auswirkungen von Rundungsfehlern zu betrachten, soll die geometrische Reihe untersucht werden:

$$\sum_{i=0}^n q^i = 1 + q + q^2 + \dots + q^n$$

Wir betrachten den Fall $q = 0.1$ und wollen die Werte von $i = 0$ bis $i = 15$ aufsummieren.

- (a) Bestimmen Sie diese Summe dadurch, dass Sie in einem Java-Programm Variablen vom Typ `double` verwenden. Benutzen Sie nur die `main`-Methode und keine weiteren eigenen oder allgemeinen Methoden.
- (b) Ändern Sie das Java-Programm so, dass Sie die gleiche Summe berechnen, aber nun in umgekehrter (absteigender) Reihenfolge, also von $i = 15$ bis $i = 0$ aufsummieren.
- (c) Das exakte Ergebnis lässt sich durch die folgende Formel beschreiben:

$$\sum_{i=0}^n q^i = \frac{1 - q^{n+1}}{1 - q}$$

Berechnen Sie dieses und vergleichen Sie es mit den Ergebnissen der beiden vorherigen Teilaufgaben.

10. Aufgabe

Eine natürliche Zahl n ist genau dann eine Primzahl, wenn sie nur durch 1 und sich selbst teilbar ist. Die Zahlen 0 und 1 sind per Definition keine Primzahlen.

- (a) Formulieren Sie einen Algorithmus in Pseudonotation, welcher für eine vorgegebene natürliche Zahl n prüft, ob n eine Primzahl ist (prim ist).
- (b) Implementieren Sie Ihren Algorithmus als einfaches Java-Programm, welches ohne weitere Funktionen/Prozeduren auskommt, wobei prinzipiell jede Zahl auf die Eigenschaft, Primzahl zu sein, getestet werden soll. Wenden Sie Ihr Programm auf die Zahl $n = 4711$ an.
- (c) Wie können Sie Ihren Algorithmus optimieren so, dass dieser mit weniger Operationen zu demselben, korrekten Ergebnis kommt?

11. Aufgabe

Implementieren Sie ein allgemein gültiges Java-Programm zur Berechnung der reellen Nullstellen der quadratischen Gleichung $a * x^2 + b * x + c = 0$. Berücksichtigen Sie dabei alle möglichen Fälle (der Parameter a, b und c). Testen Sie die Funktion mit einigen Werten, die Sie im Programm als Double „fest“ eingeben.

Hinweis: Verwenden Sie zur Lösung die Funktion Math.sqrt aus dem (implizit verfügbaren) Paket java.lang (d.h. einfach die Funktion Math.sqrt () in Ihrem Java-Programm verwenden).

Hinweis: Zum Testen, ob die Parameter a, b oder c gleich 0 sind, kann nicht der Gleichheits-Operator == verwendet werden (wegen der Rundungsfehler bzw. Ungenauigkeit). Dazu bitte eine hinreichend kleine Konstante verwenden, z.B. EPSILON = 1E-15 und den Parameter (a, b oder c) mit der Funktion Math.abs (Absolut-Betrag) auf „< EPSILON“ testen. D.h. eine Zahl wird dann als double-Null interpretiert, wenn ihr Absolutbetrag den Wert von EPSILON unterschreitet.

12. Aufgabe

Aus Java sind ja die verschiedenen Schleifen-Konstrukte while, do-while und for bekannt. Bitte stellen Sie jeweils eine beliebige for-Schleife mit Hilfe einer while-Schleife dar und umgekehrt, wobei Sie die folgende allgemeine Darstellung verwenden können:

(a) while (A) {
 B;
}

(b) for (A; B; C) {
 D;
}

13. Aufgabe

In einer Java-Anwendung kommt eine Variable vom Typ char mit dem Bezeichner zeichen vor. In Abhängigkeit des Wertes von zeichen soll das Folgende erfolgen:

Für den Wert a oder A soll das Java-Programm die Zeichenkette *Anna* ausgeben, für den Wert b oder B die Zeichenkette *Berta*. In allen anderen Fällen soll lediglich *Sonderfall* ausgegeben werden.

Hinweis: Verwenden Sie zur Realisierung die switch-Anweisung. Was passiert, wenn Sie mehrere Default-Fälle vorsehen. Welche Bedeutung hat die Position des default-Falles?

14. Aufgabe

Schreiben Sie ein Java-Programm, das für eine fest vorgegebene natürliche Zahl n testet, ob es sich um ein Schaltjahr handelt. Recherchieren Sie dazu zunächst, unter welchen Bedingungen die Jahreszahl n ein Schaltjahr darstellt.

15. Aufgabe

Schreiben Sie ein Java-Programm, das für eine fest vorgegebene positive ganze Zahl n die Quersumme ermittelt.

Hinweis: Verwenden Sie dazu nur die Operationen Division, Rest und Addition.