

**Lernziele**

- Kennenlernen der Entwicklungsumgebung
- Kennenlernen der Grundelemente einer Klasse bzw. eines Java Programms
- Nutzung der Online-Hilfe

**Aufgabe 1) Entwicklungskit**

Bevor wir uns mit komfortableren Editoren und Entwicklungsumgebungen verwöhnen, wollen wir uns zunächst mit der harten Realität der Kommandozeilen-Werkzeugen vertraut machen. Es ist war: grundsätzlich braucht man zur Java Entwicklung nicht mehr als das JDK und Notepad. Aber warum so einfach, wenn es auch umständlich geht?!

Nützliche Befehle für die Kommandozeile (DOS-Fenster)

- Aufruf des DOS-Fensters unter **Start**→**Ausführen**→**Eingabe "cmd<ENTER>"**
- Inhalt des aktuellen Verzeichnisses anzeigen (wie rechte Seite im Windows-Explorer)

**Eingabe: "dir<Enter>"**

- Wechsel zu einem anderen Laufwerk (z.B. von C: nach X:)

**Eingabe: "x:<Enter>"** (dabei kann statt x auch ein anderer Buchstabe stehen)

- Wechsel des Arbeitsverzeichnisses

**Eingabe: "cd <pfad>"**

Dabei ist <pfad> meist ein relativer Pfad, d.h. der Name eines Unterverzeichnisses (wie von "dir" angezeigt), z.B. "cd informatik". Damit kann man sich Schritt für Schritt auf dem Laufwerk bewegen - eine Ebene zurück kommt man mit "cd ..". Zur Orientierung kann man sich nach jedem Schritt das aktuelle Verzeichnis anzeigen lassen. Man kann aber auch einen absoluten Pfad (vollständiger Weg ausgehend vom Wurzelverzeichnis des Laufwerks) angeben. Dieser beginnt dann mit einem "\", z.B. "cd \informatik\termin01\aufgabe" (das ist quasi eine Anfahrtsbeschreibung: "biege vom Wurzelverzeichnis in die informatik, nimm dann termin01 und parke bei aufgabe"). Damit kann man quasi in ein beliebiges Verzeichnis springen (vorausgesetzt man kennt den genauen Ort). Das aktuelle Verzeichnis wird am Prompt (da wo der Cursor blinkt und der Rechner auf eine Eingabe wartet) angezeigt.

**Teilaufgabe a) DOS-Befehle**

Öffnen Sie ein DOS-Fenster und probieren Sie die oben beschriebenen Befehle aus. Zur Vorbereitung legen Sie in Ihrem Home-Verzeichnis auf X: bitte einen Ordner "informatik" und darunter die Unterverzeichnisse "termin01" und darin "aufgabe" an. Wechseln Sie anschließend im DOS-Fenster in dieses Verzeichnis und zeigen Sie seinen Inhalt an.

**Teilaufgabe b) Editor**

Das gute alte Notepad kennen Sie sicher – es funktioniert wie Word, nur einfacher (und öfter). Also flugs unter **Start**→**Programme**→**Zubehör**→**Notepad** aufgerufen und folgendes Java Programm eingetippt, und als "Goodbye.java" gespeichert.

```
public class Goodbye {  
  
    public static void main(String[] args) {  
        System.out.println( "Goodbye, World!");  
    }  
  
}
```

Das war doch einfach, oder? Sind sie sicher, dass Sie alle Klammern und Strichpunkte am rechten Fleck haben. Es werden noch Wetten angenommen.

### Teilaufgabe c) Der Java Compiler

Mal sehen, ob Sie wirklich alles richtig haben Der **Java Compiler** ist bei Syntaxfehlern total unbarmherzig – dafür ist er bei Funktionsfehlern stockblind (aber dazu gibt's ja Teilaufgabe c). Der Aufruf des Compilers sieht folgendermaßen aus

```
javac Goodbye.java
```

Resultat sollte eine Datei Goodbye.class sein (Kontrolle durch Eingabe von "dir"). Kommt stattdessen eine Fehlermeldung ist entweder der Pfad nicht richtig gesetzt (dann versuchen Sie es doch einfach mit "D:\Programme\Java\jdkx.y.z\bin\java Goodbye.java" – den genauen Pfad bitte im Windows-Explorer überprüfen). Kann natürlich auch sein, dass Sie sich bei der Eingabe des Programms vertippt haben. Aber der Compiler gibt ja Tipps... Wenn's dann endlich klappt, haben wir das Programm richtig geschrieben. Aber haben wir auch das richtige Programm geschrieben?

Hinweis: Der sog. Pfad ist eine Liste von Verzeichnissen, in denen der Kommandozeileninterpreter von DOS/Windows nach ausführbaren Programmen (.exe, .com, .bat, ...) sucht, die nur mit ihrem Namen referenziert werden. Was hier nicht gefunden wird, muss durch genaue (relative oder absolute) Pfadangabe referenziert werden (also im Extremfall Laufwerk+Verzeichnisse+Name). Dieses Konzept wird uns später in Java als sog. Classpath in ähnlicher Form wieder begegnen.

### Teilaufgabe d) Die virtuelle Maschine

Ta-ta! Die Nachbildung der Hardware in Software! Die Quadratur des Kreises! Nur dumm, das wir für DIE Software wieder Hardware als Ablaufumgebung brauchen. Etwas langsam ist es auch noch. Aber das überall ( "...run anywhere).

Um Ihr erstes Java Programm ablaufen zu lassen (Es ist doch Ihr Programm? Geben Sie es zu, Sie haben den Ausgabertext verändert!) genügt folgender Aufruf

```
java Goodbye
```

Die virtuelle Maschine kennt noch weitere Parameter, aber für den Anfang tut's das. Kommt raus, was Sie erwartet haben? Gut, dann haben wir das richtige Programm richtig

geschrieben. Puh! Wenn wieder ein Fehler kommt, ist abermals der Pfad schuld (s.o.). Ändern Sie nun das Programm folgendermaßen ab:

```
public class Goodbye {  
  
    public static void main(String[] args) {  
        System.out.println( "Goodbye, " + args[0] + "!");  
    }  
  
}
```

und rufen Sie es mit folgender Eingabe erneut auf

```
java Goodbye FHTE
```

Versuchen Sie statt "FHTE" noch andere Begriffe oder geben Sie mehr als einen an. Wie deuten Sie das Ergebnis (die Syntax werden wir in Kürze verstehen)?

### Aufgabe 2) Entwicklungsumgebung

Machen Sie sich nun mit der Entwicklungsumgebung (JavaEditor) vertraut, indem Sie diese aufrufen und das Programm "Goodbye" übersetzen und starten.

Dazu wählen Sie die Entwicklungsumgebung unter **Start→Programme** aus und wählen anschließend den Menüpunkt **Datei→Öffnen**. Wählen Sie nun die Datei `Goodbye.java` aus und bestätigen Sie mit OK. Zum Übersetzen der Klasse wählen Sie den Menüpunkt **Start→Compilieren**. Der Fortschritt des Übersetzungsvorgangs (inkl. etwaiger Fehlermeldungen, aber jetzt kommen ja keine mehr – es sei denn, sie bauen kurz mal wieder einen ein) wird im Statusfenster unten angezeigt. Nach erfolgreicher Übersetzung können Sie das Programm über den Menüpunkt **Start→Starte Applikation** aufrufen (In der Toolbar sind Buttons für diese Menüpunkte).

Bitte folgende Schrittfolge bei Programmänderungen beachten

1. Abspeichern aller geänderten Dateien
2. Kompilieren aller geänderten Dateien
3. Ausführen der Startklasse

→ Der Compiler und die virtuelle Maschine arbeiten mit den Daten auf der Festplatte!!!

### Aufgabe 3) Einstiegsklasse

Studieren Sie die Beispielklasse "Goodbye" und deren syntaktische Bestandteile und machen Sie sich deren Bedeutung klar:

- Welche Bedeutung hat die Methode `main` (vgl. Buch Seite xx und yy)?
-

- 
- Warum muß `main` eine Klassenmethode (d.h. `static`) sein (was ist das überhaupt)?
- 
- Warum müssen `main` und die Klasse `HelloWorld` öffentlich (`public`) sein?
- 

**Aufgabe 4) Online Dokumentation**

Öffnen Sie einen Web-Browser Ihrer Wahl und rufen Sie die Online Dokumentation auf. Versuchen Sie mit Ihrer Hilfe die Bedeutung der Befehlszeile in der Methode "main" der Beispielklasse zu klären. Beginnen Sie dazu bei der Klasse `System` und navigieren Sie über deren Klassenvariable `out` weiter zu dessen Klasse `PrintStream` und der Methode `println`. Wieviele Varianten von dieser Methode gibt es? Worin unterscheiden sich diese? Was ist Ihre naive Deutung (ohne Vorwissen) ?

---