

Java-Umgebungsvariablen

3.5.1 Setzen der Umgebungsvariablen **JAVA_HOME**

Die Umgebungsvariable JAVA_HOME wird von vielen Programmen verwendet, die auf dem JDK aufbauen, um das Installationsverzeichnis des JDK, d. h. das Verzeichnis, in dem der Java Compiler und der Java Interpreter bzw. die von der VM benötigten Bibliotheken bei der Installation abgelegt wurden, herausfinden zu können.

Die Umgebungsvariable JAVA_HOME zeigt immer auf das Basisverzeichnis des JDK, beispielsweise auf
C:\Programme\Java\jdk1.x.0_xx.

Unter Windows Betriebssystemen ist hierbei eine neue Umgebungsvariable JAVA_HOME anzulegen. Bei den aktuellen Versionen der Windows Betriebssysteme (ab Windows 7) muss in der Systemsteuerung zuerst System und Sicherheit, gefolgt von System ausgewählt werden. Anschließend ist links auf Erweiterte Systemeinstellungen zu klicken, um dann unter Umgebungsvariablen eine neue Variable für JAVA_HOME anzulegen.

Unter UNIX/Linux kann die Umgebungsvariable JAVA_HOME beispielsweise wie folgt gesetzt werden:

```
export JAVA_HOME=/usr/java/jdk1.x.0_xx
```

3.5.2 Setzen der Umgebungsvariablen **PATH**

Die PATH-Variable wird für alle Java-Tools benötigt, die im bin-Verzeichnis des JDK vorhanden sind. Dazu gehören java, javac, javah, keytool, rmic etc.

Permanente Erweiterung des Suchpfads

Um den Pfad zu den Java-Programmdateien permanent in die PATH-Variable aufzunehmen, muss diese erweitert werden. Hierfür muss – wie schon beim Setzen von JAVA_HOME – der Dialog zum Editieren von Umgebungsvariablen geöffnet werden. Dort kann unter Systemvariablen die Variable PATH ausgewählt und der Eintrag

```
;%JAVA_HOME%\bin
```

am Ende hinzugefügt werden. Durch diese Änderung steht nun der neue Wert allen Benutzern zur Verfügung. Soll die PATH-Variable nur für den aktuellen Benutzer verändert werden, so muss der Wert unter Benutzervariablen gesetzt werden.

3.5.3 Setzen der Umgebungsvariablen **CLASSPATH**

Die Umgebungsvariable CLASSPATH wird vom Compiler javac des JDK sowie vom Interpreter java des Java-Laufzeitsystems JRE benutzt, um den Weg zu benutzerdefinierten Klassen, Quellen und zu anderen erforderlichen Dateien zu finden, die nicht Teil der Java-API sind.

Wird der CLASSPATH nicht explizit gesetzt, wird das aktuelle Verzeichnis durchsucht.

Die Verzeichnisse, die die Klassen der Java-API enthalten, werden vom Compiler javac sowie vom Interpreter java grundsätzlich durchsucht. Deshalb ist es normalerweise nicht erforderlich, den CLASSPATH explizit zu setzen. Sollen jedoch Klassen oder Klassenbibliotheken benutzt werden, die sich nicht innerhalb der aktuellen Verzeichnisstruktur des aktuellen Verzeichnisses befinden, so ist es erforderlich, den CLASSPATH explizit zu setzen. Wie der CLASSPATH zu setzen ist, hängt vom jeweiligen Betriebssystem ab. Im Folgenden zwei Beispiele:

```
export CLASSPATH=. (UNIX/Linux)
```

```
set CLASSPATH=C:\wrk;. (Windows)
```

Der Punkt . hinter dem Doppel- bzw. Strichpunkt steht dabei für das aktuelle Verzeichnis.

Es ist auch möglich, mehrere verschiedene Pfade im CLASSPATH anzugeben.

Dann kann der Compiler bzw. der Interpreter nach den Klassen in verschiedenen Verzeichnissen suchen.

Der CLASSPATH kann auch direkt beim Kompilieren beziehungsweise beim Aufruf des Interpreters für ein Java-Programm angegeben werden. Das folgende Beispiel zeigt dies für das Kompilieren:

```
javac -classpath C:\wrk;. HelloWorld.java
```

Show all environment variables Commands > set

```
Path=C:\Program Files\BellSoft\LibericaJDK-17\bin\;
```

```
JAVA_HOME=C:\Program Files\BellSoft\LibericaJDK-17\
```