

Einrichten eines Packages

Zur Gruppierung und Auslagerung einzelner Klassen in einem (lokalen) Package `<packagename>` empfiehlt sich folgendes Vorgehen:

- Am Anfang jeder Datei, die eine Klasse beinhaltet, die dem Package `<packagename>` zugeordnet werden sollen, wird eine Zeile mit dem Kommando `package <packagename>;` eingefügt.
- Ein Verzeichnis mit dem Namen `<packagename>` wird als Unterverzeichnis des Verzeichnisses, in dem das Programm liegt, welches auf das Package zugreifen soll, erstellt.
- Alle Dateien, die Klassen des Packages enthalten, werden in das so generierte Verzeichnis verschoben. Alternativ kann man beim Übersetzen mit `javac` die Option `-d <pfadname>` verwenden; dies sorgt dafür, daß alle `.class` Dateien automatisch in die richtigen Unterverzeichnisse gelegt werden (falls die entsprechenden Verzeichnisse noch nicht existieren, so werden sie erzeugt).

Sollte der Name des Packages Punkte enthalten (z.B. bei Umsetzung des von SUN vorgeschlagenen Konzepts, immer den umgekehrten Domainnamen voranzustellen), so führt jeder Punkt zu einem neuen Unterverzeichnis für die Lokalisierung der Package Dateien. Beispielsweise führt ein Name `de.tzi.example` dazu, daß die hierzugehörigen Java Klassendateien in dem Unterverzeichnis `de/tzi/example` zu finden sind.

Um **statische** Methoden aus dem so erzeugten Package aufzurufen, sollte ein Programm folgende Regeln befolgen:

- Das Package sollte vor der eigentlichen Klassendefinition importiert werden mit dem Kommando `import <packagename>.*;`
- Eine statische Methode wird dann über `<classname>.<methodname>(<parameterlist>);` aufgerufen, wobei `classname` eine Klasse aus dem entsprechenden Package bezeichnet.

Damit die notwendigen `.class` Dateien aus einem Package gefunden werden, muß das aufrufende Programm an oberster Stelle derartiger Verzeichnisstrukturen ausgeführt werden, d. h. in dem Beispiel müßte es ein Unterverzeichnisstruktur des aktuellen Verzeichnisses namens `de/tzi/example` geben. Alternativ muß das oberste Verzeichnis dem `CLASSPATH` hinzugefügt werden (siehe hierzu `man javac` für ausführlichere Informationen, oder Balzert „Lehrbuch Grundlagen der Informatik“, S. 356).

Alternativ lassen sich die Methoden eines Packages auch ohne `import` Anweisung verwenden, wenn immer der Name des Packages vorangestellt wird, also dann über `<packagename>.<classname>.<methodname>(<parameterlist>);`