

Die Klasse `IOTools` – Tastatureingaben in Java

Kurzbeschreibung

Java ist eine komplexe Sprache, und ihre Klassenbibliothek hat einen enormen Umfang. Viele Funktionalitäten wurden von den Entwicklern abstrahiert und in allgemeiner Form implementiert, sodass insbesondere Anfänger auf große Schwierigkeiten stoßen. Leider ist die Eingabe eine davon.

Ein Programm erhält die Eingaben der Benutzerinnen und Benutzer über sogenannte *Eingabeströme* (englisch: input streams). Man kann sich diese Ströme am besten wie einen altmodischen Nachrichtenticker vorstellen. Auf einem schmalen Streifen kommen Nachrichten an, Zeichen für Zeichen aneinander gereiht. Welche Daten und Informationen auf diesen Streifen stehen, ist dem Gerät dabei egal – es handelt sich nur um eine Ansammlung von Zeichen. Es obliegt der Leserin bzw. dem Leser, die Streifen an der richtigen Stelle abzureißen und die Daten auf dem Streifen zu interpretieren.

So oder so ähnlich können wir uns auch die Ströme in Java veranschaulichen. Ein Strom besteht aus nichts weiter als einer Ansammlung von Bits, die vom Programm in irgendeiner Form interpretiert werden müssen. Hierbei kann die Quelle dieser Zeichen verschieden sein: eine Datei auf der Festplatte, ein Dokument aus dem Internet oder eben die Eingabe von der Tastatur. Die zur Verfügung stehenden Methodenaufrufe sind in allen Fällen gleich.

Haben wir es jedoch geschafft, dem Computer mit unserem Programm klarzumachen, dass wir als Eingabestrom die Tastatur verwenden wollen, sind wir noch lange nicht am Ziel. Die vordefinierten Methoden empfangen nämlich in der Regel eine Zeichenkette – einen `String`. Wir wollen im Allgemeinen jedoch keine Strings, sondern beispielsweise `int`-Werte, `double`-Zahlen oder eventuell einzelne Zeichen. Zwar gibt es in der Klassenbibliothek wiederum Möglichkeiten, diese aus unserem `String` zu extrahieren; hierfür benötigen wir jedoch meistens Wissen, das über den Wissensstand einer Programmieranfängerin oder eines Programmieranfängers hinausgeht. Um hier Abhilfe zu schaffen, wurden die `IOTools` geschrieben.

Anwendung der `IOTools`-Methoden

Um die `IOTools` in Ihren Programmen verwenden zu können, müssen Sie sich natürlich zunächst einmal das entsprechende Paket (die `Prog1Tools`) besorgen und es auf Ihrem Rechner installieren. Auf der Website www.grundkurs-java.de steht alles, was Sie dazu benötigen, zum Download bereit. Halten Sie sich dabei genau an die Installationsanweisung.

Um die Klasse `IOTools` in Ihre Programme einzubinden, müssen Sie an den Anfang jeder Klasse, in der die `IOTools`-Methoden eingesetzt werden sollen, die Zeile

```
import Prog1Tools.IOTools;
```

setzen. Diese Zeile veranlasst den Übersetzer, die Klasse `IOTools` aus dem Paket `Prog1Tools` einzubinden.

Folgende Methoden sind unter anderem definiert:

- Die Methode `readInteger` (wie auch ihre Kurzform `readInt`) liest eine Zahl vom Typ `int` von der Tastatur ein und gibt diese als Ergebnis zurück. Um beispielsweise zwei ganze Zahlen von der Tastatur einzulesen und in den Variablen `a` und `b` zu sichern, genügt folgendes Programmstück:

```
int a = IOTools.readInteger();  
int b = IOTools.readInteger();
```

- Die Methode `readDouble` liest eine Zahl vom Typ `double` ein. Obiges Beispiel würde also für `double`-Zahlen wie folgt aussehen:

```
double a = IOTools.readDouble();  
double b = IOTools.readDouble();
```

- Die Methode `readLong` liest eine Zahl vom Typ `long` ein. Die Methoden `readShort` und `readFloat` tun dies für die Datentypen `short` und `float`.
- Die Methode `readLine` liest eine ganze Textzeile (abgeschlossen durch den Druck auf die Eingabetaste).
- Die Methode `readString` liest ein einzelnes „Textwort“ von der Tastatur. Ein Textwort besteht aus einem String, der weder durch Leer- noch Tabulator- noch Zeilenendezeichen auseinandergerissen ist. Geben wir beispielsweise die Zeile

```
_____ Konsole _____  
Dies ist eine schoene Zeile.
```

ein und lesen diese im Programm mit `readString` ein, erhalten wir lediglich `Dies` als Ergebnis. Um das nächste Wort zu lesen, muss die Methode erneut aufgerufen werden.

- Die Methode `readChar` liest ein einzelnes Zeichen, das nicht gleich dem Leerzeichen, Zeilenendezeichen oder dem Tabulatorzeichen ist. Die Methode basiert hierbei auf der `readString`-Methode, das heißt, es werden Textworte eingelesen und in ihre einzelnen Komponenten aufgespalten. Das Programmstück

```
IOTools.readChar();  
char a = IOTools.readChar();  
int b = IOTools.readInteger();  
double c = IOTools.readDouble();
```

liefert bei der einzeiligen Eingabe

Konsole

```
abc123  456  5.73
```

also `a='b'`, `b=456` und `c=5.73`, da die Ziffern 123 noch zum ersten Textwort gehören.

- Die Methode `readBoolean` liest einen `boolean`-Wert ein. Hierbei ist auf Groß- und Kleinschreibung zu achten; die Eingabe `True` codiert beispielsweise *keinen* Wert vom Typ `boolean`. Es muss vielmehr `true` heißen.

Wie wir in obigen Beispielen gesehen haben, kann auch mehr als eine einzulesende Information pro Zeile eingegeben werden (man muss sie lediglich durch Leerzeichen trennen). Hierbei muss man natürlich auf die Reihenfolge der Eingaben achten. Der Befehl `readInteger` wird bei der Eingabe

Konsole

```
Ich gebe jetzt einmal 13 ein.
```

als Ergebnis den Wert 13 zurückgeben, da dies die erste gültige Ganzzahl ist. Die davor stehenden Textworte werden verworfen und überlesen.

Bei Konsoleneingaben ist es wichtig, vor jeder Eingabe zumindest eine kurze Information darüber auszugeben, dass nun eine Eingabe erfolgen soll. Man sollte also stets mit einem solchen **Prompt** (deutsch: Aufforderung) arbeiten, um dem Anwender bzw. der Anwenderin des Programms zu verstehen zu geben, dass das Programm nun auf eine Eingabe wartet.

Die `IOTools` unterstützen diesen Mechanismus, indem die Klasse alle bereits genannten Methoden auch in einer Variante mit zusätzlichem Parameter vom Typ `String` zur Verfügung stellt. Über diesen kann der `readXYZ`-Methode eine Zeichenkette übergeben werden, die unmittelbar vor der Eingabe auf die Konsole ausgegeben wird. Man spart sich auf diese Weise entsprechende Ausgabeanweisungen, etwa mit `System.out.print`.

Wenn wir also beispielsweise mit den Anweisungen

```
int a = IOTools.readInteger("Geben Sie den Wert a ein: ");
double b = IOTools.readDouble("b = ");
```

arbeiten, könnte ein Programmablauf etwa so aussehen:

Konsole

```
Geben Sie den Wert a ein: 1234
b = 12.45e7
```

Weitere Beispielanwendungen der `IOTools` finden sich in den Programmen `IOToolsTest` und `IOToolsDemo`.