

Aufgabe 9: Scannen von Zahlen

Implementieren Sie die Methode `readNumber(t)` gemäß ihrer Spezifikation in Kapitel 2. Die Methode wird von `next` aufgerufen, um eine ganze Zahl zu erkennen. Die Tokenart `number` soll in `t.kind` gespeichert werden und der Wert der Zahl in `t.numVal`. Wenn die Zahl zu groß ist, um in einer `int`-Variablen Platz zu haben, soll eine Fehlermeldung ausgegeben werden.

Lösung

Beim Aufruf von `readNumber` enthält `ch` die erste Ziffer der Zahl. `readNumber` muss nun weitere Ziffern lesen und sie in einem `char`-Array `lex` zwischenspeichern, um schließlich den Ziffernstring in eine Zahl zu konvertieren und diese in `t.numVal` zu speichern. Wenn die Zahl zu groß ist, wird eine Fehlermeldung ausgegeben, aber `t.kind` wird trotzdem auf `number` gesetzt, um den Parser nicht zu "verwirren".

```
private static char[] lex = new char[128];

private static void readNumber (Token t) {
    int i = 0;
    do {
        lex[i++] = ch; nextCh();
    } while ('0' <= ch && ch <= '9');
    t.kind = number;
    t.val = new String(lex, 0, i);
    try {
        t.numVal = Integer.parseInt(t.val);
    } catch (NumberFormatException e) {
        t.numVal = 0; Parser.error("number too large");
    }
}
```

Der Scanner benutzt hier die Methode `Parser.error`, die allgemein verwendet wird, um Fehlermeldungen samt Zeilen- und Spaltennummer des aktuellen Tokens auszugeben. Solange der Parser noch nicht existiert, können Fehlermeldungen einfach auf die Konsole ausgegeben werden.