

## Aufgabe 10: Scannen von Zeichenkonstanten

Implementieren Sie die Methode `readCharCon(t)` gemäß ihrer Spezifikation in Kapitel 2. Die Methode wird von `next` aufgerufen, um eine Zeichenkonstante (z.B. `'x'`) zu erkennen. Die Tokenart `charCon` soll in `t.kind` gespeichert werden und der numerische Wert der Zeichenkonstanten in `t.numVal`. Wenn die Zeichenkonstante fehlerhaft ist (z.B. `"`, `'abc'`), soll eine Fehlermeldung ausgegeben werden.

### Lösung

Beim Aufruf von `readCharCon` enthält `ch` das öffnende Hochkomma. `readCharCon` muss nun bis zum schließenden Hochkomma (oder bis zum Zeilenende) weiterlesen und den Inhalt der Zeichenkonstanten in einem `char`-Array `lex` zwischenspeichern. Anschließend wird dieser Inhalt analysiert und ihr numerischer Zeichenwert in `t.numVal` gespeichert. Wenn die Zeichenkonstante fehlerhaft ist, wird eine Fehlermeldung ausgegeben, aber `t.kind` wird trotzdem auf `charCon` gesetzt, um den Parser nicht zu "verwirren".

```
private static char[] lex = new char[128];
private static void readCharCon (Token t) {
    int i = 0;
    t.kind = charCon;
    nextCh();
    while (ch != '\"' && ch != '\n' && ch != eofCh) { // read character constant
        lex[i++] = ch; nextCh();
    }
    if (ch == '\n' || ch == eofCh) {
        Parser.error("invalid character constant");
    } else if (i == 1 && lex[0] != '\\') { // analyze character constant
        t.numVal = lex[0];
    } else if (i == 2 && lex[0] == '\\') {
        if (lex[1] == 'r') t.numVal = '\r';
        else if (lex[1] == 'n') t.numVal = '\n';
        else if (lex[1] == 't') t.numVal = '\t';
        else Parser.error("invalid character constant");
    } else Parser.error("invalid character constant");
    if (ch == '\"') nextCh();
}
```

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.