



TokenDefinition

Die Definition der Token erfolgt in Form von regulären Ausdrücken. Der Aufbau der Datei orientiert sich dabei an der Lex-Language.

RegexToDfaConverter

Diese Komponente sorgt für den Aufbau eines DFA aus gegebenen regulären Ausdrücken. Der Weg vom regulären Ausdruck hin zum DFA wird durch die Subkomponenten Regex2NFA, NFA2DFA und MinimizeDFA gegangen.

RegexToNfa

Diese Subkomponente erledigt den Aufbau des NFA aus gegebenen regulären Ausdrücken.

NfaToDfa

Diese Subkomponente konvertiert den gegebenen NFA in einen Dfa über die Einführung von Mengenzuständen.

MinimizeDfa

Minimiert den gegebenen Dfa mittels Markierungsalgorithmus basierent auf dem Satz von Myhill-Nerode.

DfaProvider

Der DfaProvider stellt den DFA für den TokenMatcher bereit. Die Erzeugung des DFA delegiert der DfaProvider an den RegexToDfaConverter. Den erhaltenen DFA speichert der DfaProvider zunächst als persistentes Objekt für die spätere Wiederverwendung. Sofern solch ein persistentes Objekt zu einer bekannten Sprache bereits vorhanden ist, wird anstelle der erneuten Erzeugung eines DFA das gespeicherte Objekt geladen und bereitgestellt.

TokenMatcher

Liest den Quellcode zeichenweise über die Lookahead-Schnittstelle und ändert entsprechend den aktuellen Zustand im DFA. Liefert den Tokenstrom der erkannten Token.

BufferedReader

Liest den Quellcode in ein Pufferpaar mit Wächtern. Liefert auf Anfrage das nächste Zeichen im Lookahead oder ermöglicht gegebenenfalls auch Rückschritte.

