

VCC: Entwicklung eines einfachen Compilers

In diesem Beispiel soll ein einfacher Compiler entwickelt werden, der primitive mathematische Ausdrücke als Eingabedatei lesen und berechnen kann und anschließend das Rechenergebnis in die Ausgabedatei schreibt.

VCC präsentiert sich nach dem Start wie in Abbildung 1 gezeigt. Zunächst sollte auf einem Blatt Papier eine Grammatik aufgestellt werden, die anschließend in VCC eingegeben werden kann. Für das Taschenrechnerbeispiel sollen die vier Grundrechenarten und Klammerausdrücke verwendet werden. Um dieses Beispiel nicht unnötig zu komplizieren, werden ausschließlich ganze Zahlen verwendet.

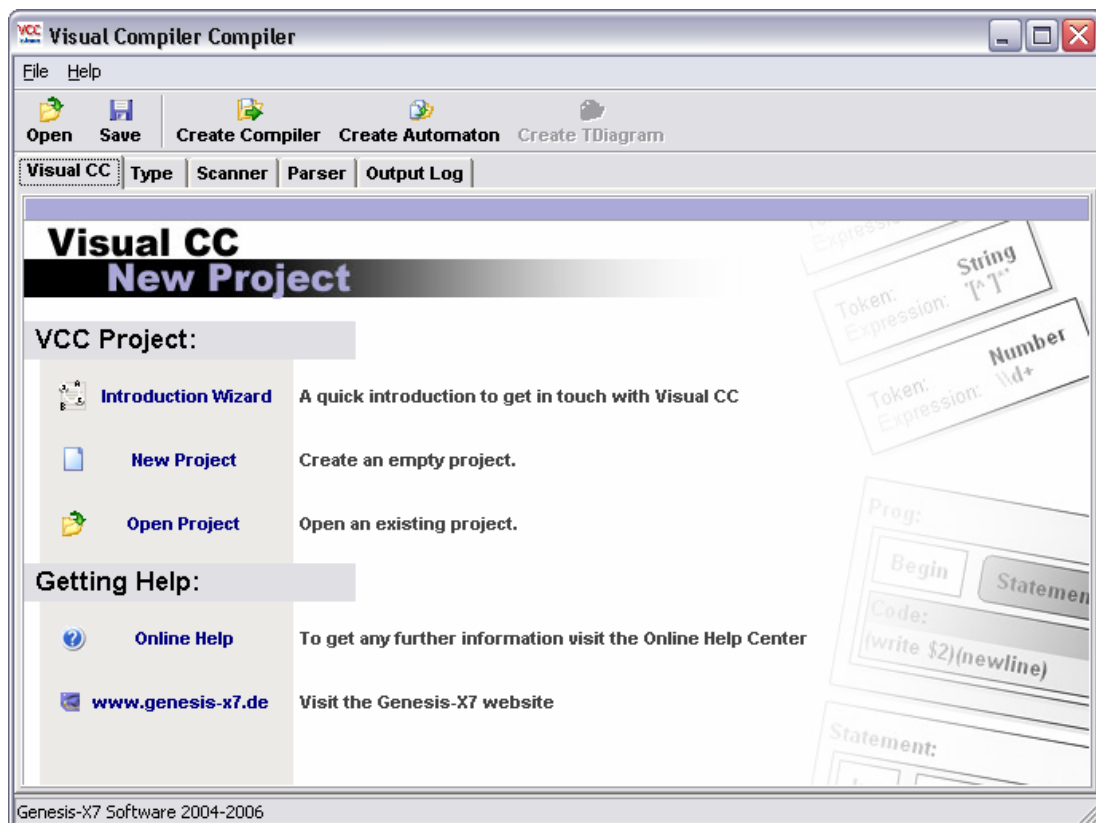


Abbildung 1: VCC Startbildschirm

Die im Weiteren verwendete Grammatik:

Rechnung	→	Ausdruck Rechnung ϵ
Ausdruck	→	Ausdruck * Ausdruck Ausdruck / Ausdruck Ausdruck + Ausdruck Ausdruck - Ausdruck (Ausdruck) Zahl
Zahl	→	Ziffer Zahl ϵ
Ziffer	→	1 2 3 4 5 6 7 8 9 0

Der Scanner für diese Grammatik beinhaltet nur 7 Token. In Abbildung 2 ist auszugsweise ein Scanner für obige Grammatik gezeigt. Wichtig ist bei den Expressions (reguläre Ausdrücke), entsprechende Sonderzeichen mit einem vorangestellten „\“ zu ergänzen. Das Nichtterminal Zahl wurde bereits im Scanner durch die Verwendung von „[0-9]+“ aufgelöst. Dies ist nicht zwingend nötig, spart aber Aufwand.

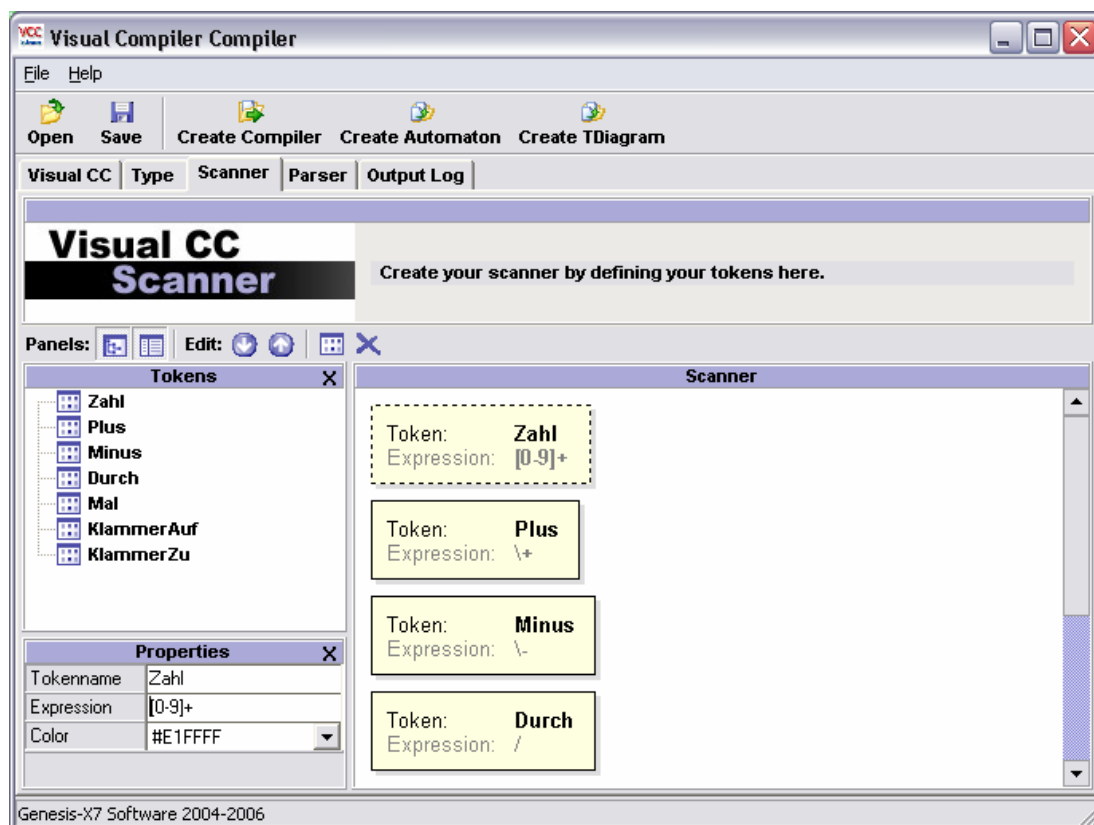


Abbildung 2: VCC Scanner

Anschließend kann über die Tabs auf die Parseransicht gewechselt werden. Zunächst werden die Nichtterminale als Regeln (Rechnung, Ausdruck und Zahl) hinzugefügt.

Wie in Abbildung 3 gezeigt, werden diese 3 Regeln in VCC dargestellt. Nun kann Schritt für Schritt die gesamte Grammatik in diese Darstellung übertragen werden. Dazu dienen die Schaltflächen im oberen Bereich, oder es kann mit dem Popup Menü (durch Rechtsklick) gearbeitet werden. Unter Token bzw. Regeln werden ausschließlich die bereits vorhandenen in Scanner bzw. Parser vorgegeben.

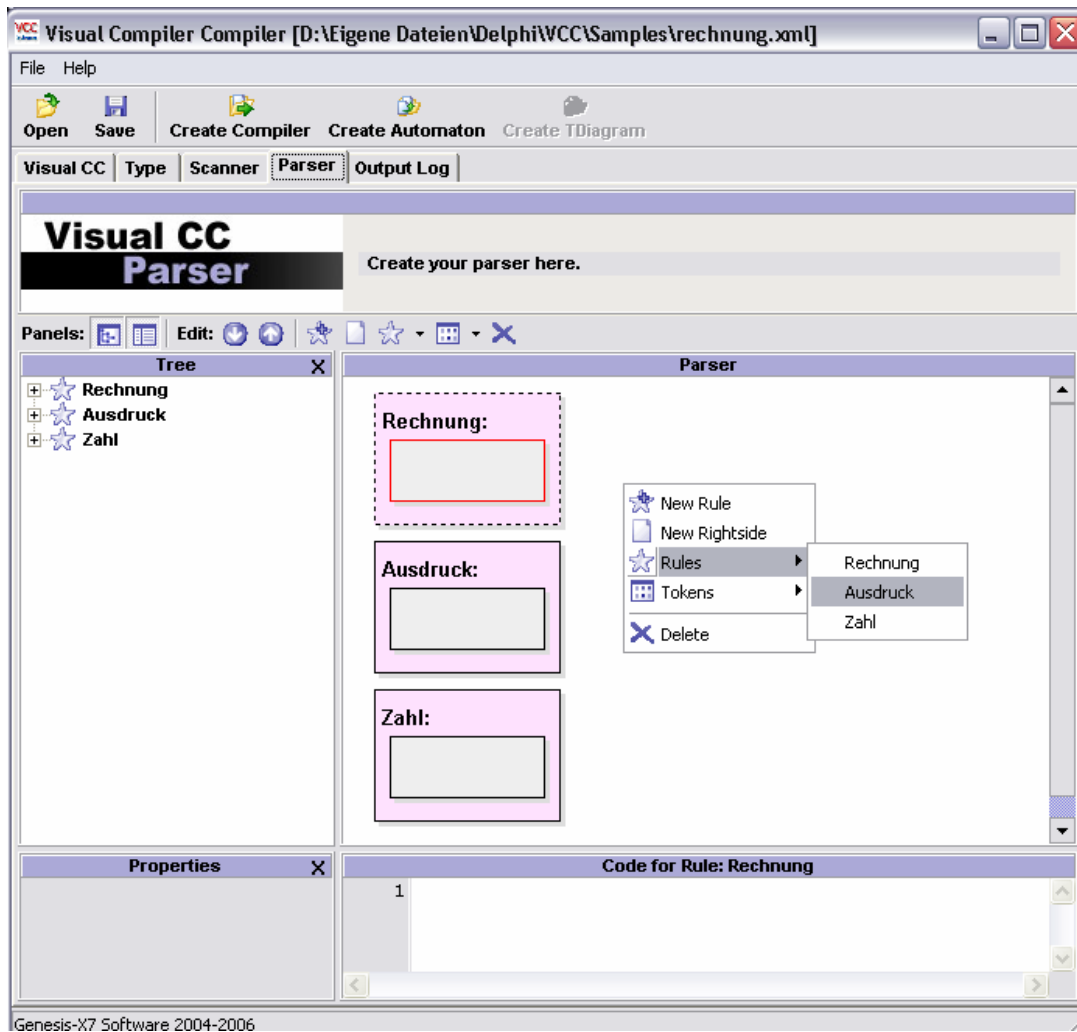


Abbildung 3: VCC Parser 1

Wie die einzelnen Regeln in VCC übertragen werden, soll am Beispiel des Startsymbols „Rechnung“ gezeigt werden.

Rechnung \rightarrow Ausdruck Rechnung | ϵ

Der linke Teil „Rechnung“ wird durch die Regel selbst repräsentiert. Für diese Regel gibt es zwei mögliche Auflösungen. Einmal ein „Ausdruck“ gefolgt von einer neuen „Rechnung“, oder Epsilon. Die typische Darstellung „|“ wird in VCC durch weiße Rechtecke innerhalb der Regeln untereinander dargestellt. Über „New Rightside“ können weitere Rechtecke hinzugefügt werden. In das erste bereits vorhandene Rechteck (Selektion durch einfachen Mausklick, erkennbar durch rote Umrandung) werden nun zunächst die Regel „Ausdruck“ und anschließend die Regel „Rechnung“ aufgenommen. Durch einen Klick auf „New Rightside“ wird die zweite, in diesem Fall leere rechte Regelseite hinzugefügt. Das Ergebnis wird in Abbildung 4 gezeigt.

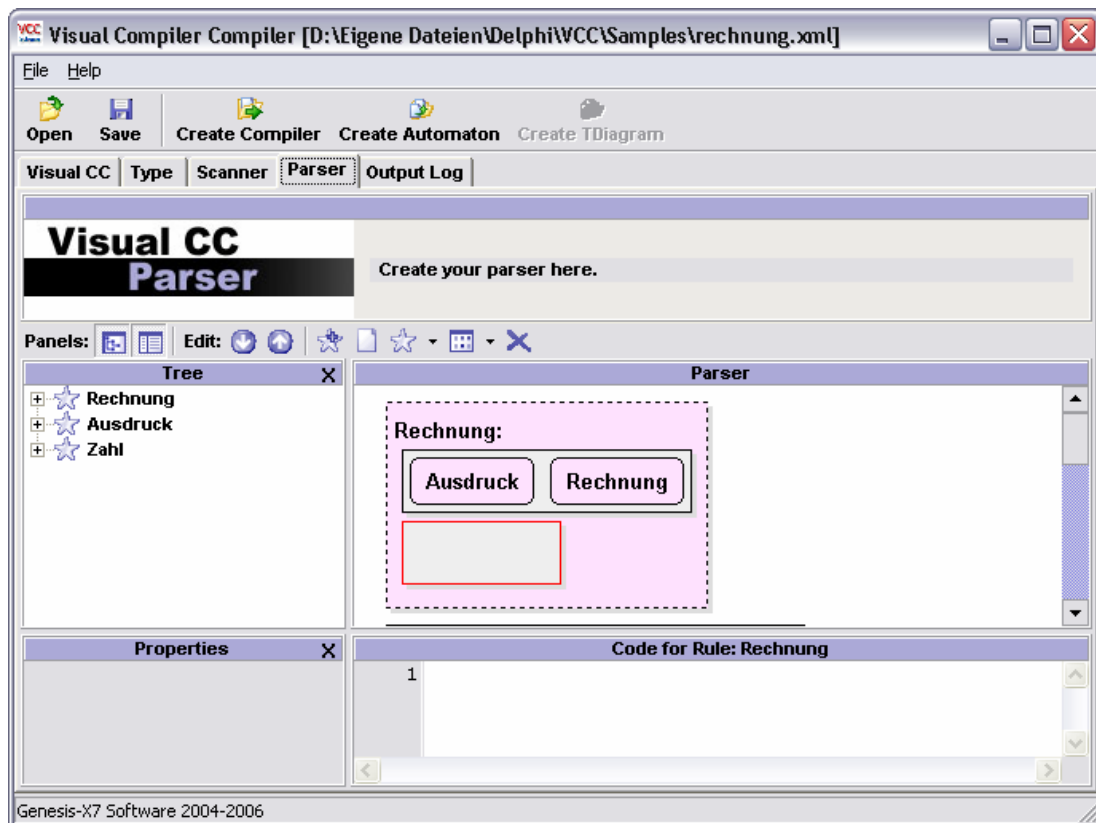


Abbildung 4: VCC Parser 2

Analog kann die Regel „Ausdruck“ übertragen werden, wobei hier auch Token (für die Rechenzeichen) verwendet werden (Abbildung 5).

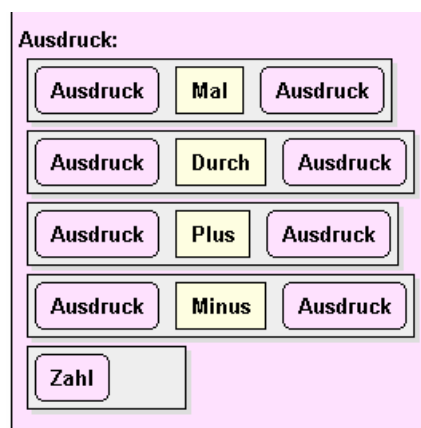


Abbildung 5: VCC Parser 3

Wurde die gesamte Grammatik auf diese Weise in VCC übertragen, kann die Codegeneration des Compilers hinzugefügt werden. Dabei müssen äquivalent zu YACC für die einzelnen Regeln Codefragmente hinzugefügt werden, die bei der Abarbeitung der Regel ausgeführt werden sollen. Wichtig ist hierbei das Verständnis der Abarbeitungsreihenfolge und die Kenntnis der Sondervariablen \$\$ und \$n. Hierzu kann zum einen das mitgelieferte VCC

Beispielprojekt mit Hinweisen für diese Problematik herangezogen werden, oder aber auch Literatur zu YACC verwendet werden.

Eine typische Codezeile für das Taschenrechnerbeispiel (hier in C#) würde wie folgt aussehen:

```
$$ = (int.Parse($1) + int.Parse($3)).ToString();
```

Da VCC prinzipiell immer mit Zeichenketten in den \$\$, \$n Variablen arbeitet, müssen diese zunächst in Ganzzahlen umgewandelt, anschließend addiert und danach wieder in einen String zurück transformiert werden.

In Abbildung 6 ist dieses Quellcodefragment bei der Additionsregel in VCC eingetragen (unterer Bereich). Analog werden die anderen Regeln mit Quellcode ergänzt (In diesem Falle muss lediglich das Rechenzeichen abgeändert werden.).

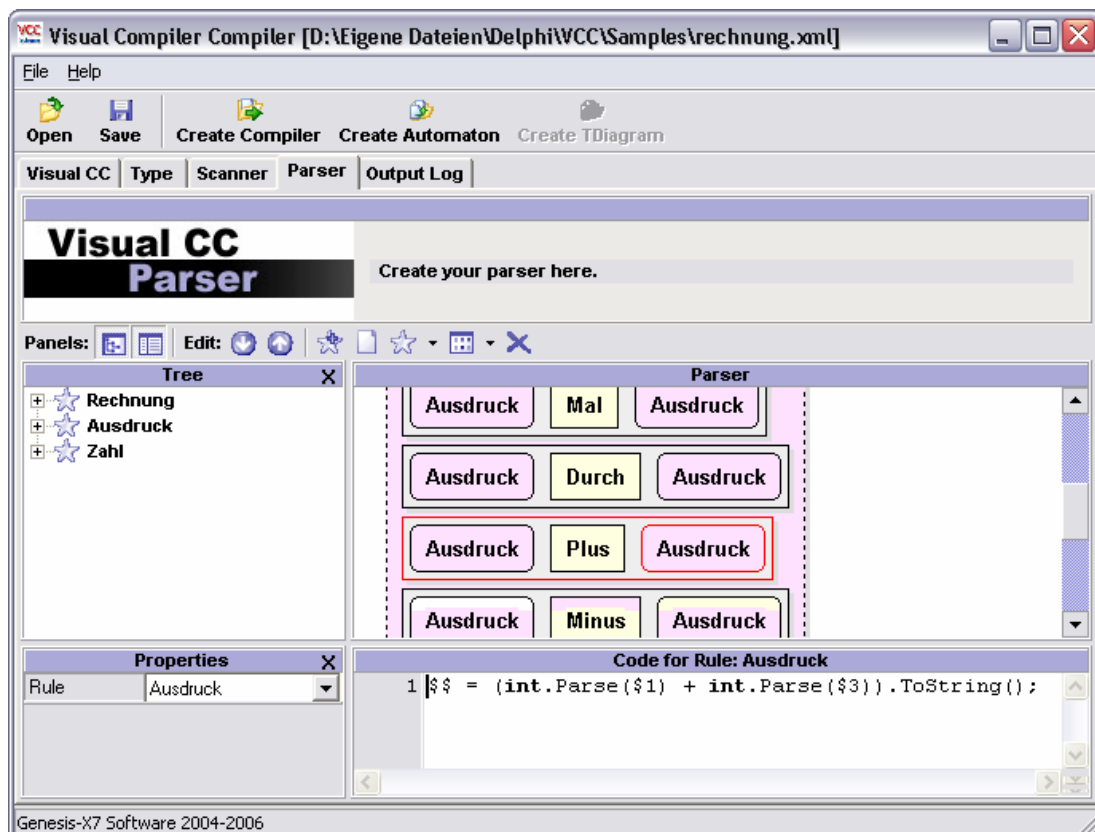


Abbildung 6: VCC Parser Code

Über „Create Compiler“ kann ein fertiger Compiler erzeugt werden. VCC öffnet automatisch eine Konsole im betreffenden Verzeichnis und gibt Hinweise über die Startparameter:

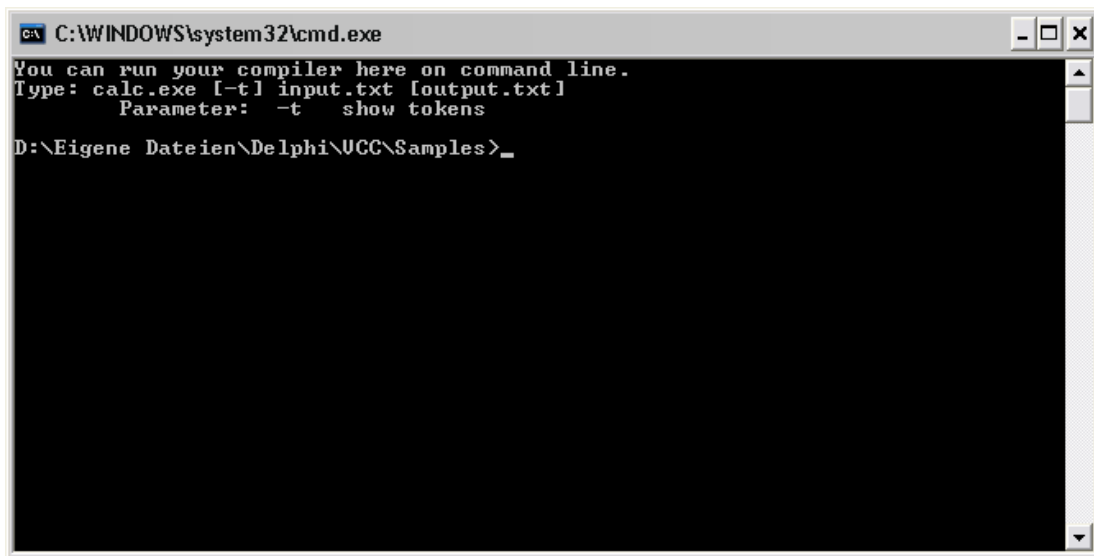


Abbildung 7: VCC Konsole

Eine Eingabedatei für den entwickelten Taschenrechner-Compiler könnte nun wie folgt aussehen:

```
((32*12)-(82/2))-123
```

Über den Aufruf `calc.exe inputdatei.txt` wird dieser Ausdruck berechnet und das Ergebnis: 220 zurückgegeben.

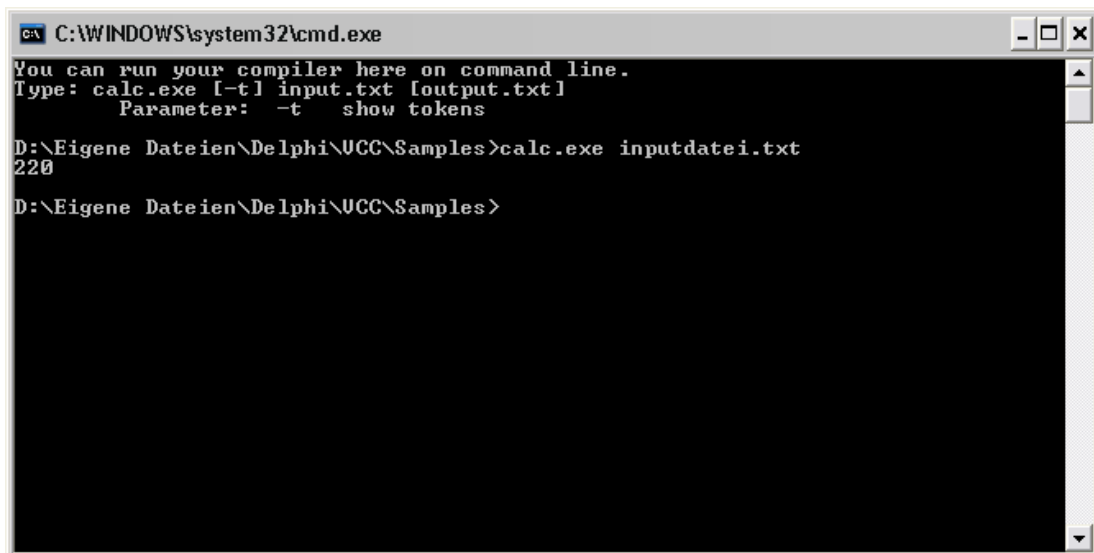


Abbildung 8: VCC Konsole Ausgabe