

Debugger

Beim Programmieren können drei Arten von Fehlern auftreten:

- der **Syntaxfehler**, wird vom Compiler entdeckt und verhindert die erfolgreiche Erstellung des ausführbaren Programs (program.exe).
- der **Laufzeitfehler**, der vom Compiler nicht bemerkt wird, beim Programmablauf in einer bestimmten Situation aber eine Fehlermeldung hervorruft. Passiert zum Beispiel, wenn man in einem Array mit 5 Zellen auf die Zelle 10 zugreifen möchte.
- der **logische Fehler**, der weder beim Kompilieren noch zur Laufzeit zu einer Fehlermeldung führt; allerdings tut das Programm nicht das, was sich der Programmierer gedacht hatte. Dieser Art von Fehler können Sie mit dem Debugger auf die Spur kommen.

In obiger Reihenfolge steigt auch die Schwierigkeit, den Fehler zu entdecken. Bei einem Syntaxfehler wird die fehlerhafte Zeile vom Compiler normalerweise markiert und eine Meldung angezeigt.

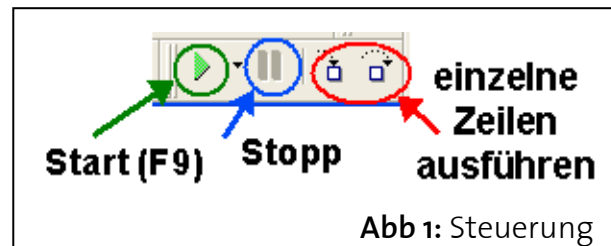
Bei einem Programmabbruch zur Laufzeit (**Laufzeitfehler**) wird häufig auch die fehlerhafte Zeile angezeigt, sofern das Programm aus der Delphi-Programmierungsumgebung (IDE, integrated development environment) heraus gestartet worden ist.

Logische Fehler sind all jene, bei denen Sie nicht wissen, warum das Programm nicht das tut, was es tun sollte. Z.B. kann beim Klicken auf einen Knopf etwas passieren, was nicht geschehen sollte oder es passiert gar nichts, das Resultat einer Berechnung kann falsch sein, eine Anweisung wird nie ausgeführt usw. Bei diesen Fehlern müssen wir uns selber an der Nase nehmen, denn der Computer macht genau das was er sollte, nämlich was wir programmiert haben.

Bei der Suche nach den letzten beiden Fehlerarten hilft uns der in der Delphi-IDE integrierte **Debugger**. Wir können den Programmablauf an einer bestimmten Stelle im Quellcode anhalten und von da an schrittweise verfolgen, wohin die Programmausführung springt, welche Routinen durchlaufen werden, welche Werte Variablen haben usw.

Programmausführung

Bis jetzt haben Sie Ihre Programme entweder mit F9 oder mit dem grünen Startpfeil von Delphi gestartet (siehe Abb1). Hier werden Sie nun neben dieser Möglichkeit noch weitere kennen lernen, die Ihnen helfen den Programmablauf zu verfolgen.



F9 (Start): Die Anwendung wird kompiliert und gestartet. Dabei werden nur Module kompiliert, die sich seit dem letzten Kompilieren geändert haben. Anschließend wird die Anwendung neu gelinkt. Werden Syntaxfehler gefunden, werden diese angezeigt und der Vorgang abgebrochen.

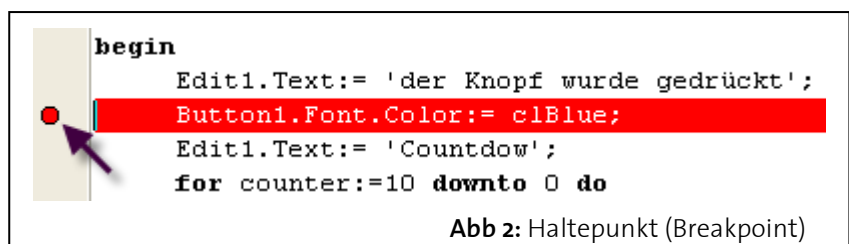
F8: Das Programm wird Zeile für Zeile ausgeführt. Prozeduren- und Funktionsaufrufe werden dabei in einem Schritt abgearbeitet (übersprungen). Sollen auch diese zeilenweise ausgeführt werden, muss stattdessen die Funktion "Einzelne Anweisung" (F7) verwendet werden. Soll das Programm normal weiterlaufen, muss F9 gedrückt werden.

F7: Das Programm wird Zeile für Zeile ausgeführt. Bei Aufrufen von Prozeduren und Funktionen wird in diese verzweigt und auch diese zeilenweise abgearbeitet. Soll das Programm normal weiterlaufen, muss F9 gedrückt werden.

F4: Das Programm wird bis zu der Stelle normal ausgeführt, an der der Cursor im Code-Editor steht. Dann wird angehalten. Von dieser Stelle an kann dann zeilenweise vorgegangen werden (F7 oder F8) oder das Programm normal weiter ausgeführt werden (F9).

Haltepunkte (Breakpoints)

Bei Haltepunkten handelt es sich um **Markierungen**, bei denen der **Programmablauf angehalten** wird. Sie können zum Beispiel in einer Schleife einen Haltepunkt setzen, um zu sehen, ob die Schleife überhaupt ausgeführt wird. Sie können aber auch die erste Zeile einer Funktion oder Prozedur markieren, um zu sehen, ob diese auch wirklich einmal ausgeführt wird.



Setzen von Haltepunkten

Haltepunkte werden gesetzt indem man den linken Rand (grau) mit der Maus anklickt. Es erscheint dann ein roter Punkt.

Stoppen bei Haltepunkten

Wird bei der Ausführung des Programms ein Haltepunkt erreicht, dann stoppt das Programm und zeigt wieder die Programmierumgebung an. Ein grüner Pfeil zeigt an, bei welchem Haltepunkt die Ausführung gestoppt wurde. Sie können jetzt aber nichts am Quellcode verändern, da dies Einfluss auf die weitere Ausführung hätte. Diese kann aber nicht mehr verändert werden, da das Programm ja bereits kompiliert wurde. Wenn Sie jetzt etwas ändern möchten müssen Sie die Ausführung abbrechen.

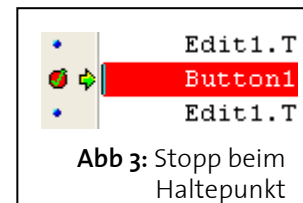


Abb 3: Stopp beim Haltepunkt

Von Haltepunkt zu Haltepunkt

Wenn Sie von einem Haltepunkt zum nächsten springen möchten drücken Sie F9. Möchten Sie nach dem Anhalten zeilenweise weiterfahren wählen Sie F8 oder F7.

Untersuchen von Variablen

Wenn die Programmausführung bei einer Zeile stehen bleibt (Haltepunkt, zeilenweise Ausführung mit F7 oder F8), dann können die Variablen untersucht werden. Meist interessiert der Wert, den die Variable hat.

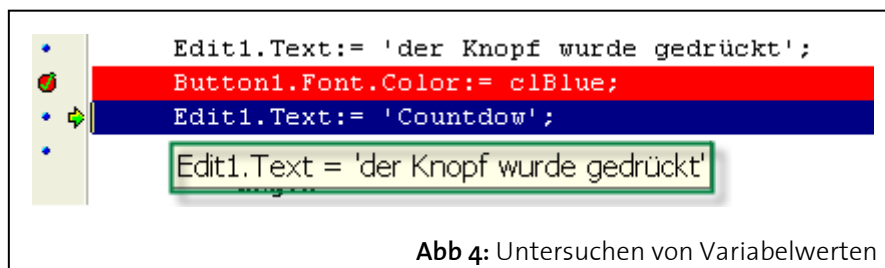


Abb 4: Untersuchen von Variabelwerten

Hier sehen Sie ein Beispiel für eine Wertanzeige. In der ersten Zeile wurde in das Textfeld Edit1 der Text ‚der Knopf wurde gedrückt‘ geschrieben. Zwei Zeilen weiter wird in dieses Textfeld ein neuer Text geschrieben. Stoppt man die Ausführung des Programms nun bei dieser Zeile (grüner Pfeil), dann kann man mit der Maus über die Zeile fahren. Bei ‚Text‘ wird dann die grün eingerahmte Information angezeigt. Sie sehen, dass der Wert noch nicht verändert wurde, da die dritte Zeile noch nicht ausgeführt wurde.

Und nun sind Sie an der Reihe!

- Versuchen Sie doch bei ihrem nächsten Programm einmal die zeilenweise Ausführung.
- Setzen Sie ein paar Haltepunkte und schauen Sie, wann diese erreicht werden.
- Untersuchen Sie die Werte der Variablen, indem Sie mit der Maus über die Zeile fahren (geht aber nur während der Ausführung).