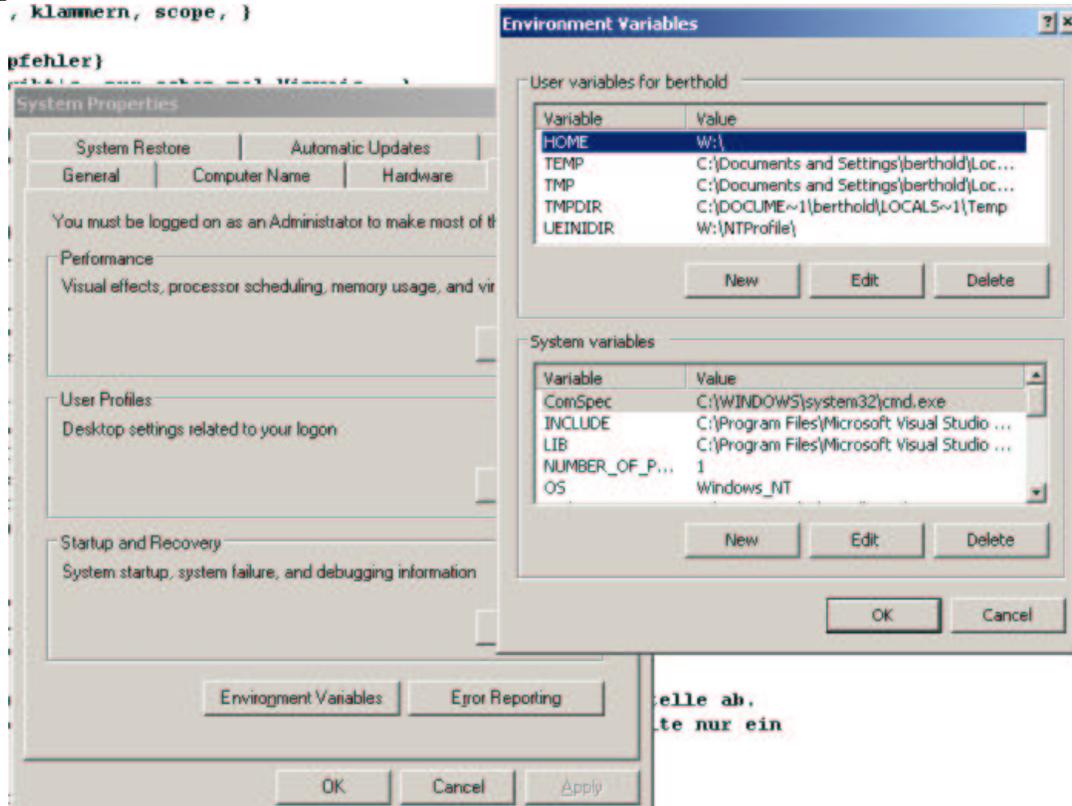


1 IDE-Ersatz: Konfiguration eines Editors (Beispiel)

1.1 Ultraedit

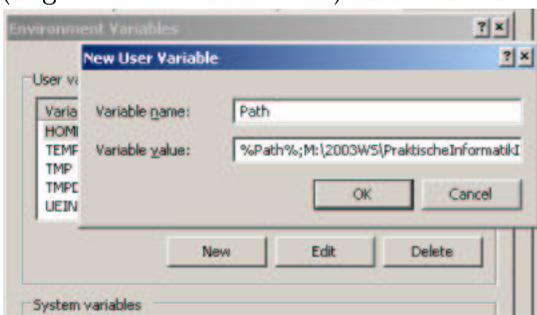
Der Editor *UltraEdit*, der am Fachbereich auf Windows-Rechnern verfügbar ist, lässt sich je Benutzer individuell konfigurieren, wenn man vorher die Variable `UEINIDIR` auf ein global sichtbares Verzeichnis mit Schreibrechten setzt.

Zum Setzen dieser Variable wählt man im “Control Panel” (Startmenü) die Option *Performance and Maintenance* → *System*. Im Register “Advanced” findet sich “Environment Variables”, die man z.B. wie folgt modifiziert:



UltraEdit legt nun seine Konfiguration *uedit32.ini* an dieser Stelle ab. Dadurch werden Änderungen an der Konfiguration möglich und wirken sich auf allen Rechnern des Fachbereichs aus. Wohin die Variable zeigt, ist Geschmacksache, es sollte nur ein *global sichtbares* Benutzerverzeichnis sein (und natürlich mit Schreibrechten :-).

Wo wir gerade hier sind, setzen wir am besten noch einen Pfad auf das Hugs-Verzeichnis (und evtl. das des GHC, Unterverzeichnis `\bin`). Eine neue User-Variable `PATH` erhält mit `%Path%` den bisherigen Pfad, dahinter (abgetrennt mit Semikolon) zusätzlich für Hugs den Eintrag `M:\2003WS\PraktischeInformatikIII\Hugs98`:

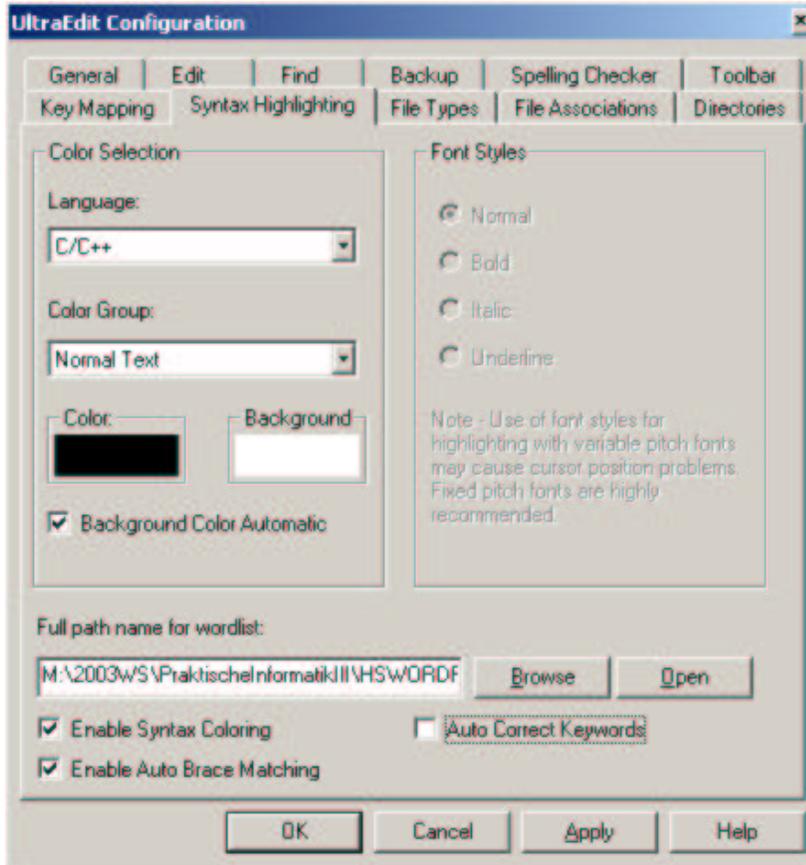


Achtung! Wer die bisherige Konfiguration (Java-Knöpfe etc.) behalten will, muss *jetzt* die lokale Datei `C:\Program Files\UltraEdit\Uedit32.ini` an die neue Stelle kopieren, bevor Ultraedit gestartet wird.

1.1.1 Syntaxhervorhebung

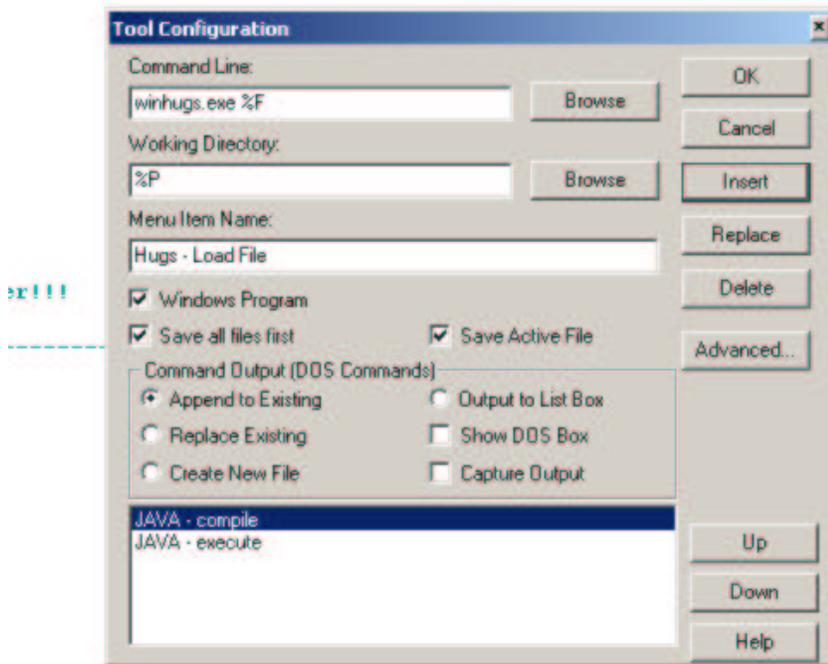
UltraEdit kann automatisch die Haskell-Syntax hervorheben, wie man es von IDEs für andere Sprachen (BlueJ, JBuilder, Visual Studio) gewohnt ist.

Dafür muss die Sprache Haskell in seinem sog. *wordfile* definiert sein. Diese Arbeit muss man sich allerdings nicht selbst machen, ein passender Abschnitt kann heruntergeladen und manuell in die Datei eingefügt werden. In *M:\2003WS\PraktischeInformatikIII* liegt eine vorbereitete Datei *HSWORDFILE.TXT*, die man ebenfalls benutzen kann. Die gewünschte Datei wird unter *Advanced* → *Configuration* auf der Seite “Syntax Highlighting” angegeben:



1.1.2 HUGS-”Bedienknopf”

Wie die bereits vordefinierten Werkzeuge *Java - compile* und *Java - execute* kann in UltraEdit über *Advanced* → *Tool Configuration* ein Bedienelement definiert werden, mit dem die aktuelle Datei in Hugs geladen wird. Das folgende Bild zeigt die dafür vorzunehmenden Einstellungen:



Mit einem Klick auf "Insert" wird das Bedienelement definiert und kann (nach einem Klick mit der rechten Maustaste auf die Bedienleiste und Auswahl "Customize") der Bedienleiste hinzugefügt werden.

1.2 Emacs

Für Linux-Benutzer steht ein Emacs-Mode zur Verfügung, der eine Syntaxhervorhebung für Haskell-Programme definiert. Daneben kann er auch z.B. automatisch Funktionstypen anzeigen und sinnvolle Einrückungen vorschlagen. Außerdem können Hugs und GHC direkt aus Emacs aufgerufen werden.

Der Haskell-Mode kann unter <http://www.haskell.org/haskell-mode/haskell-mode-1.44.tar.gz> heruntergeladen werden. Seine Installation ist im Archiv ausführlich beschrieben; sie erfolgt durch Anpassung der Datei `$HOME/.emacs` mit den folgenden Zeilen:

```
(setq load-path (cons "/hier/archiv/ausgepackt" load-path))
(setq auto-mode-alist
      (append auto-mode-alist
              '(("\\. [hg]s$" . haskell-mode)
                ("\\. hi$" . haskell-mode)
                ("\\. l[hg]s$" . literate-haskell-mode))))
(autoload 'haskell-mode "haskell-mode"
          "Major mode for editing Haskell scripts." t)
(autoload 'literate-haskell-mode "haskell-mode"
          "Major mode for editing literate Haskell scripts." t)
(add-hook 'haskell-mode-hook 'turn-on-haskell-font-lock)
(add-hook 'haskell-mode-hook 'turn-on-haskell-decl-scan)
(add-hook 'haskell-mode-hook 'turn-on-haskell-doc-mode)
(add-hook 'haskell-mode-hook 'turn-on-haskell-indent)
; (add-hook 'haskell-mode-hook 'turn-on-haskell-simple-indent)
(add-hook 'haskell-mode-hook 'turn-on-haskell-hugs)
```

1.3 Andere Werkzeuge

Für andere Editoren stehen ebenfalls Erweiterungen für Haskell zur Verfügung, es gibt (angeblich) auch richtige IDEs. Unter <http://www.haskell.org/libraries/#programDevelop> findet sich eine ganze Sammlung, wo sich jede/r sein Lieblingswerkzeug aussuchen kann.