

Java-Editor

Integrierte Java-Entwicklungsumgebungen setzen hohe Anforderungen an die Ausstattung der Rechner. Diese Anforderungen sind in der Schule nicht immer umzusetzen. Sollen die Schüler auch zu Hause mit diesen Entwicklungssystemen arbeiten, so kommen eigentlich nur frei verfügbare Systeme in Frage. Da bleiben kaum noch Systeme übrig, die in der Schule tatsächlich einsetzbar sind. Dies war für mich Anlass, den nachfolgend beschriebenen Java-Editor zu entwickeln.

Download


Aktuell steht die [Version 7.03a vom 17.06.2006](#) zum Download bereit.

Sprachen

Der Java-Editor kann mittels Sprachdateien auf verschiedene Sprachen eingestellt werden. Derzeit stehen zur Verfügung

 [Deutsch](#)

 [English](#)

 [Español](#) by Héctor Fabián Echeverri Quintero

 [Dutch](#) by Pieter van der Hoeven

Installation

Sie wird unter [Installation](#) beschrieben.

FAQ

Antworten zu einigen E-Mail-Anfragen finden Sie unter [FAQ](#).

Versionen

Ich habe einen Überblick über die veröffentlichten [Versionen](#) zusammen gestellt.

Newsletter

Wenn Sie über neue Versionen des Java-Editors per Newsletter unterrichtet werden wollen, so melden Sie sich einfach an:

Ihre E-Mail-Adresse:

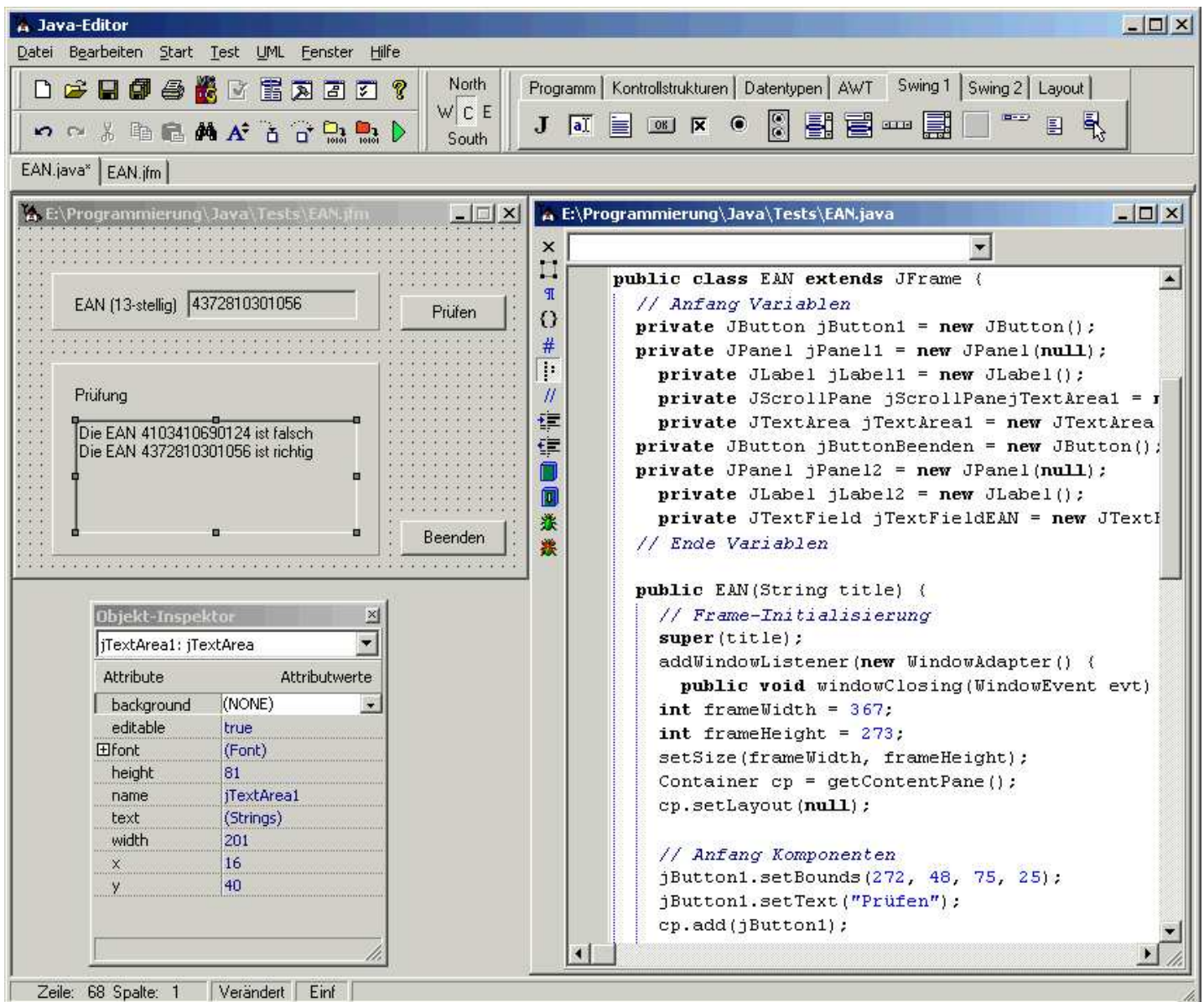
Anmelden Abmelden

Überblick

Im Bild sehen Sie das Programmfenster des Java-Editors mit Menü- und Symbolleiste, sowie einigen Registern mit Hilfen zur Programmentwicklung. Die Ausstattung ist an den Notwendigkeiten der Schule ausgerichtet. Daher gibt es zum Beispiel das Register *Programm* mit den Programm-Grundstrukturen für Konsolen-Anwendungen, Applikationen und Applets und das Register *Kontrollstrukturen* mit den Java-Kontrollstrukturen.

GUI-Designer

Ab Version 5.0 steht auch ein GUI-Designer im Java-Editor zur Verfügung. Die bisherigen Erfahrungen im Unterricht zeigen, dass das Erstellen von grafischen Benutzeroberflächen für die Schülerinnen und Schüler ziemlich kompliziert ist. Der richtige Umgang mit Layout-Managern will gelernt sein. Mit dem BorderLayout-Manager sein Layout wunschgemäß zu erzeugen kann eine sehr zeitintensive Angelegenheit sein. Der einfachere Zugang mit einem FlowLayout-Manager ergibt oft nicht das gewünschte Layout. Der im Java-Editor verfügbare GUI-Designer benutzt absolute Positionierung, womit man ziemlich schnell sein Layout gestalten kann, und das ist genau das, was in der Schule gebraucht wird.



Zum Anlegen eines GUI-Formulars klicken Sie auf der Registerkarte "Programm" auf eines der vier Symbole: Frame, JFrame, Applet oder JApplet.

Zum Platzieren einer GUI-Komponente klickt man zunächst auf das entsprechende Symbol der Registerkarten AWT, Swing1 bzw. Swing2 und dann auf das GUI-Formular. Anschließend kann man die Komponente dort positionieren und mit dem Objekt-Inspektor konfigurieren. Im Quelltext werden die jeweiligen Einstellungen synchron vorgenommen. Dazu werden die Abschnitte

```
// Anfang Variablen
// Ende Variablen
```

und

```
// Anfang Komponenten
// Ende Komponenten
```

benutzt. Diese Abschnittskennzeichnungen und der darin befindliche, automatisch erzeugte Quelltext darf nur auf eigene Gefahr geändert werden.

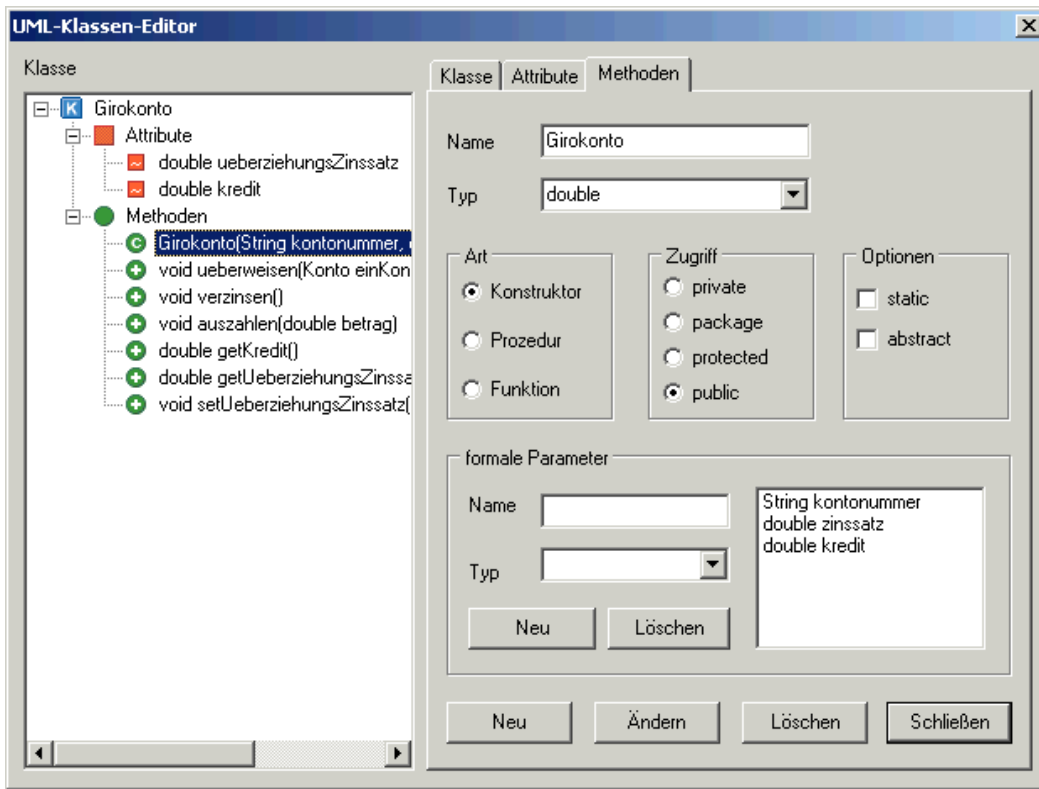
Über das Symbol  eines Quelltext-Fensters können Sie das zugehörige GUI-Formular öffnen und zusammen mit dem Quelltext-Fenster anordnen. Schließt man das GUI-Formular funktionieren die Komponenten-Symbole der Registerkarten AWT, Swing1 und Swing2 wie bisher.


Das GUI-Formular verfügt über ein Kontextmenü zur Bearbeitung der Komponenten.

In anderen, insbesondere kommerziellen Entwicklungssystemen wird die Struktur der grafischen Benutzeroberfläche als Baum dargestellt. Der Java-Editor verzichtet auf ein weiteres Fenster und zeigt die Struktur im Quelltext durch entsprechendes Einrücken der zugehörigen Variablen im Variablenabschnitt an.

UML-Klassen-Editor

Mit dem UML-Klassen-Editor können Sie neue Klassen erzeugen und bestehende Klassen bearbeiten. Er liest Java-Dateien und stellt sie als Klasse mit Attributen und Methoden dar. Änderungen im UML-Klassen-Editor werden direkt in die zugehörige Java-Datei übernommen.



Den UML-Klassen-Editor rufen Sie über das Symbol  im Editorfenster auf. Sie können auch im UML-Diagramm die gewünschte Klasse doppelklicken, um den Klassen-Editor aufzurufen.

Datei-Menü

Im Datei-Menü steht neben den üblichen Operationen wie Neu, Öffnen, Speichern und Drucken noch das Exportieren von Java-Programmen im HTML- oder RTF-Format zur Verfügung. So kann die Syntax-Auszeichnung im Editor-Fenster problemlos in Dokumentationen der Schüler bzw. in Arbeitsblätter des Lehrers übernommen werden.

Bearbeiten-Menü

Der Editor unterstützt die Arbeit mit der Windows-Zwischenablage, das Suchen und Ersetzen und mehrstufiges Rückgängig machen von Editoroperationen. Markierte Textblöcke lassen sich ein- und ausrücken sowie ein- und auskommentieren, markierter Text kann im Text-, HTML- und RTF-Format in die Windows-Zwischenablage kopiert werden.

Start-Menü

Über das Start-Menü bzw. die zugehörigen Symbole kann das im aktiven Editor-Fenster stehende Programm kompiliert werden. Fehlermeldungen werden im Compiler-Fenster angezeigt. Per Doppelklick auf eine Fehlermeldung wird der Cursor an die Fehlerstelle im Quelltext positioniert. Natürlich können Programme auch gestartet werden.

Applets werden im Appletviewer angezeigt. Die zum Aufruf nötigen HTML-Dokumente werden automatisch oder manuell per Befehl generiert. Startet man hingegen das zum Applet gehörige HTML-Dokument, so wird das Applet im Browser angezeigt.

Zusätzlich lassen sich über das Start-Menü der Debugger, Disassembler, Dokumentierer, Archivierer und Stil-Prüfer aufrufen.

Test-Menü

Mit den Befehlen des Test-Menüs führen Sie ein Programm unter der Kontrolle des Debuggers schrittweise aus, setzen oder löschen Haltepunkte, werten Ausdrücke aus und bearbeiten die Liste der zu überwachenden Ausdrücke.

Wenn Sie im Fenster *Attribute* oder *Parameter und Lokale Variablen* auf ein Objekt doppelklicken so werden Objektdetails angezeigt.

UML-Menü

Das UML-Fenster können Sie ein- und ausschalten. Zum Speichern und Laden von Klassendiagrammen benutzen Sie die normalen Öffnen und Speichern-Symbole und wählen dabei den Dateityp '.uml' aus. Sie können ein Klassendiagramm als Bild speichern oder in die Zwischenablage zur direkten Weiterverarbeitung kopieren. Mit dem Befehl 'Ordner öffnen' können Sie aus allen Java-Dateien eines Ordners sowie seiner Unterordner ein Klassendiagramm erstellen.

Das UML-Fenster bietet eine eigene Symbolleiste zur Bearbeitung des Diagramms an. Aus den geöffneten Java-Dateien können die Klassen und Klassenbeziehungen eingelesen werden. Klassenbeziehungen lassen sich per Doppelklick bearbeiten. Neue Klassen können angelegt werden. Zum Anlegen einer neuen Klassenbeziehung markieren Sie die Startklasse. Dann ziehen Sie mit der Maus eine Verbindung zur Zielklasse.

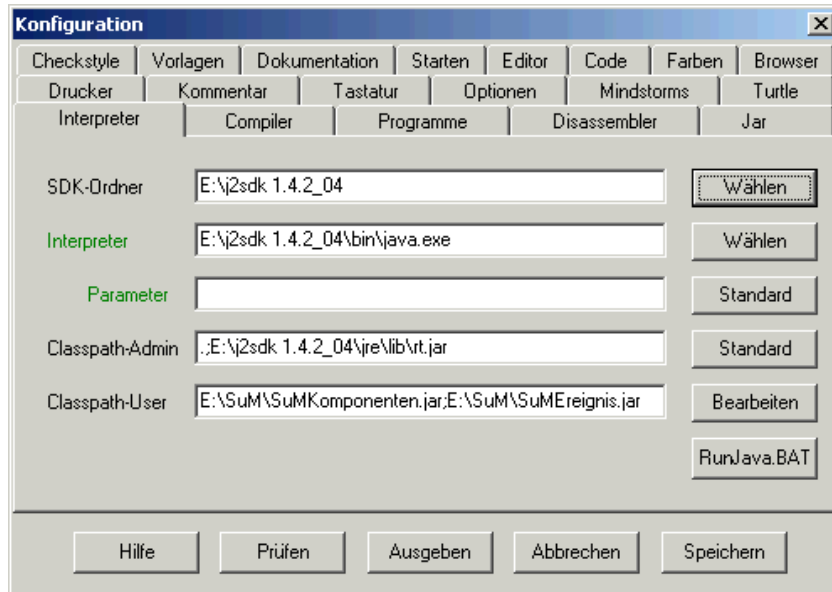
Fenster-Menü

Neben den Grundfunktionen zum Anordnen, Öffnen und Schließen der Fenster kann die Schriftart ausgewählt und der Konfigurationsdialog aufgerufen werden. Im Folgenden werden die verschiedenen Registerkarten des Konfigurationsdialogs beschrieben.

Interpreter

Einzustellen ist der Pfad zum Java-SDK. Das macht man am besten über den Wählen-Schalter, denn dann werden die anderen Pfade entsprechend angepasst. Ausgehend von diesem Pfad werden die davon abhängigen Unterverzeichnisse, Programme und Dokumentationen ermittelt.

Mittels *Prüfen* können die gesetzten Verzeichnisse und Dateien kontrolliert werden. Rot unterlegte Dateien und Verzeichnisse fehlen.



Der Administrator kann unter Classpath-Admin eine feste Vorgabe machen, zusätzlich hat der Benutzer mit Classpath-User die Möglichkeit den vorgegebenen Classpath zu ergänzen.

Bei den Classpath-Einträgen ist Vorsicht bei der Schreibweise geboten. Der Classpath-Admin sollte stets mit dem Punkt als Stellvertreter für das aktuelle Verzeichnis beginnen. Darauf folgend können weitere Pfade oder Jar-Dateien angegeben werden. Die einzelnen Angaben sind ohne Zwischenraum mit einem Semikolon voneinander zu trennen. Der Eintrag `E:\j2sdk1.4.2_04\jre\lib\rt.jar` wird für den Jikes-Compiler gebraucht.

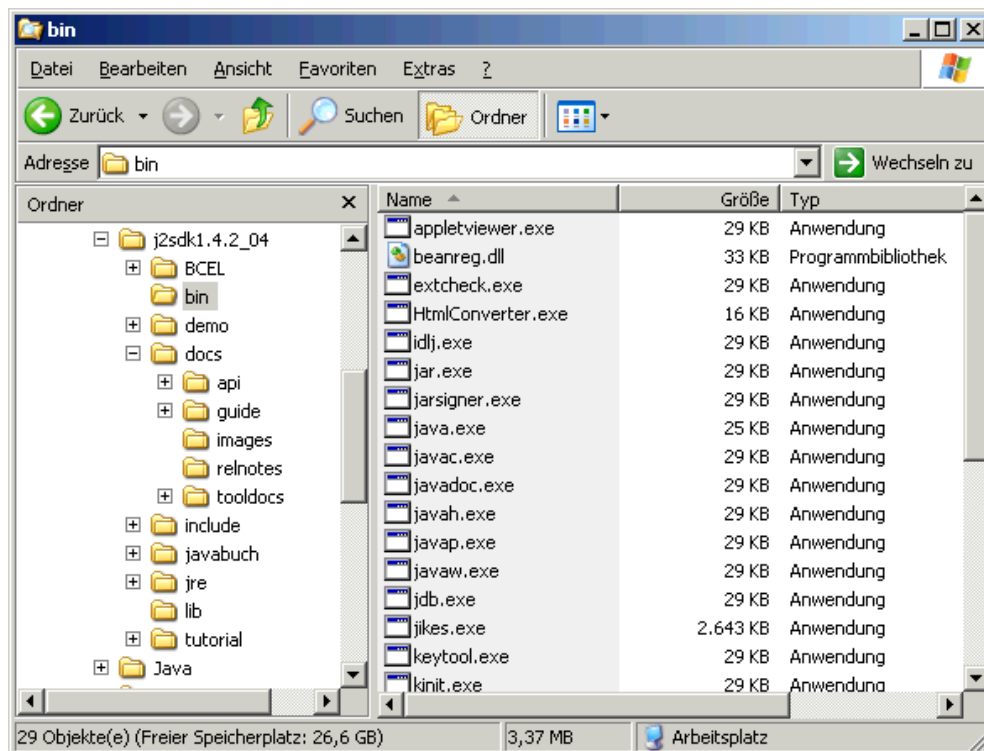
Zur Einbindung des Paketes "Stifte und Mäuse" vom Bildungsserver Nordrhein-Westfalen könnte der Classpath-User wie folgt gesetzt werden:

```
E:\SuM\SuMKomponenten.jar;E:\SuM\SuMEreignis.jar
```

Sollen zusätzliche Pakete wie im Javabuch beschrieben installiert werden, so legt man diese in einem Ordner `E:\Java\classes` ab und erweitert den Classpath-Admin dann so:

```
.;E:\j2sdk1.4.2_04\jre\lib\rt.jar;E:\Java\classes
```

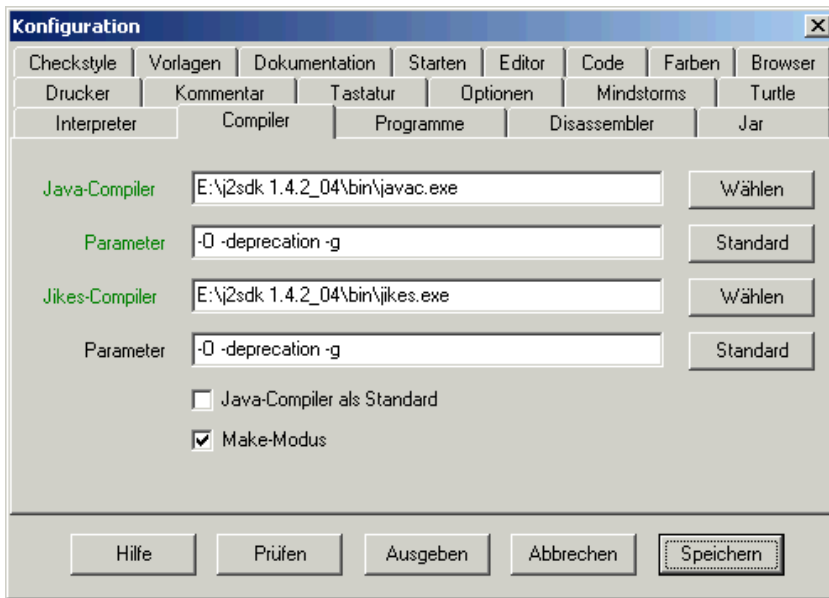
Gemäß der Standardinstallation geht der Java-Editor von folgender Verzeichnisstruktur aus:



Im Unterverzeichnis *bin* werden also die Java-Werkzeuge wie Compiler `javac.exe`, Interpreter `java.exe` und Dokumentierer `javadoc.exe` erwartet. Da die Dokumentation separat von Sun angeboten wird, ist darauf zu achten, dass diese in das Unterverzeichnis *docs* installiert wird.

Compiler

Hier stellen Sie die Pfade für die beiden Compiler `javac.exe` und `jikes.exe`, sowie die zugehörigen Compilerparameter ein. Mit der Option "Java-Compiler als Standard" legen Sie für die Menübefehle "Compilieren" und "Alle compilieren" den Compiler fest.



Im Make-Modus werden beim Compilieren des Hauptprogrammes zunächst alle anderen geänderten Dateien gespeichert. Dann wird bei allen geöffneten Dateien geprüft, ob die java-Datei jünger als die zugehörige class-Datei ist. Ist die java-Datei jünger, so wird sie frisch compiliert. Zum Schluss wird dann das Hauptprogramm compiliert.

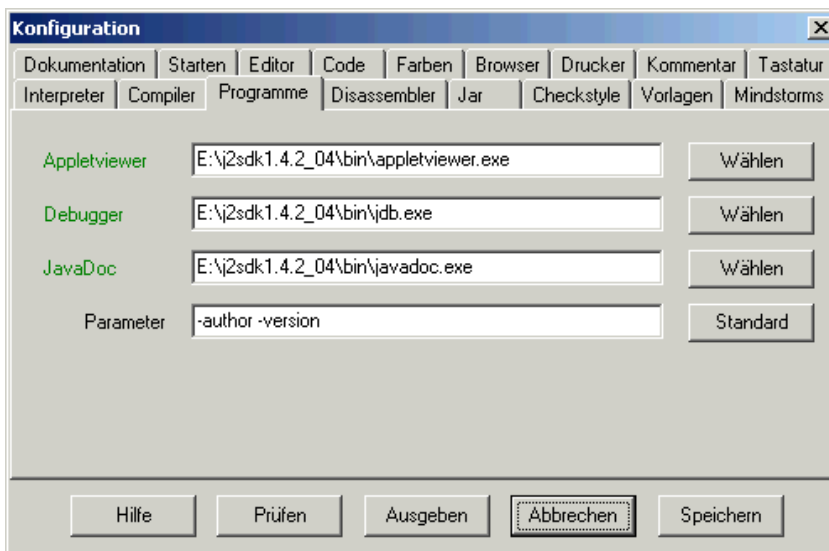
Jikes-Compiler

Als Alternative zum Java-Compiler von Sun bietet sich der deutlich schnellere Compiler » **Jikes von IBM** an. Kopieren Sie "jikes.exe" in den Ordner \bin des Java-SDK, also in den Ordner des Java-Compilers. Wählen Sie dann jikes.exe als Compiler aus.

Der Jikes-Compiler kommt nicht mit Umlauten in Datei- oder Ordernamen zurecht. Zudem ist er nicht in der Lage, debugfähigen Code zu erzeugen. Arbeitet man mit jikes als Standard-Compiler so wird deshalb automatisch vor dem Debuggen der Quelltext nochmal mit dem javac-Compiler kompiliert.

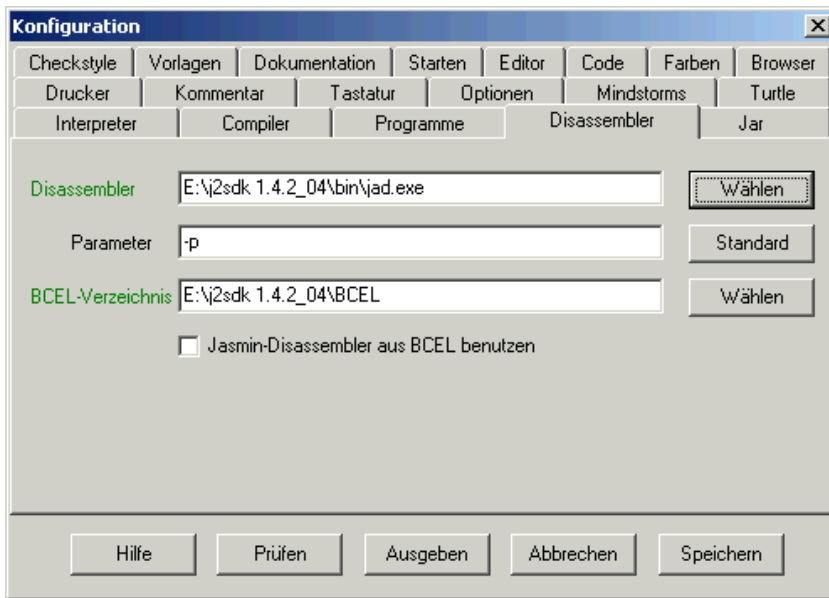
Programme

Hier werden die vom Java-Editor unterstützten Java-Tools konfiguriert:



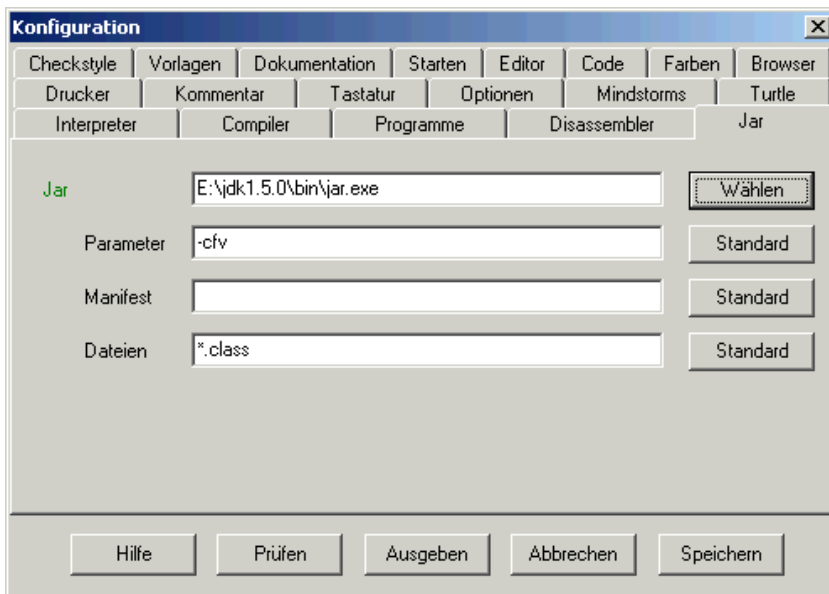
Disassembler

Sie können standardmäßig den SUN-Disassembler, den empfehlenswerten » **JAD-Disassembler** oder den » **Jasmin-Disassembler** verwenden.



Jar

Auf dieser Registerkarte legen Sie die gewünschten Parameter für die Funktion "Erzeugen" des Archivierers jar fest.



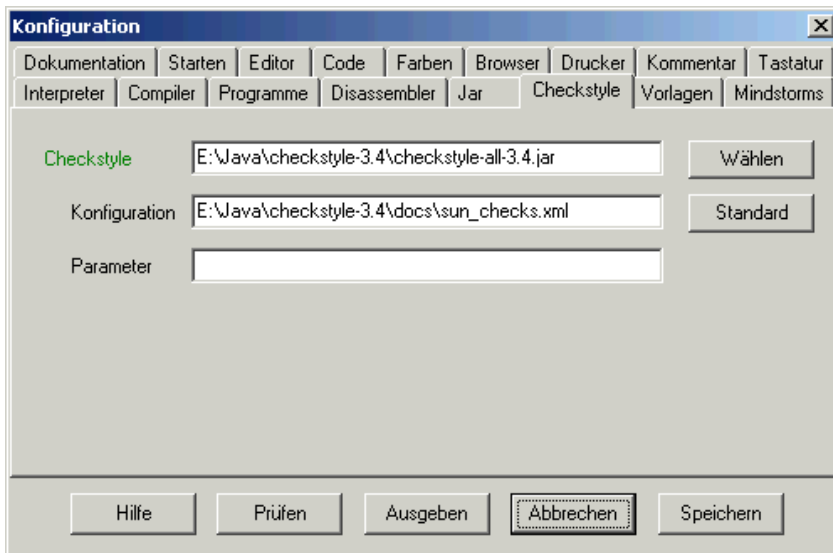
Das Start-Menü enthält fünf jar-Operationen:

- Erzeugen: Erzeugt gemäß den in der Registerkarte eingestellten Optionen eine jar-Datei.
- Packen: Packt alle java- und jfm-Dateien im Ordner des aktuellen Programms in ein jar-Archiv.
- Inhalt anzeigen: Zeigt den Inhalt eines jar-Archivs an.
- Entpacken: Entpackt ein Jar-Archiv.
- Öffnen: Öffnet ein Jar-Archiv im Java-Editor

Checkstyle

Checkstyle ist ein Entwicklungswerkzeug, das einem Programmierer hilft Programme nach vorgegebenen Programmierrichtlinien zu schreiben.

Die Download-Adresse des Checkstyle-Paket ist: >> <http://checkstyle.sourceforge.net/>

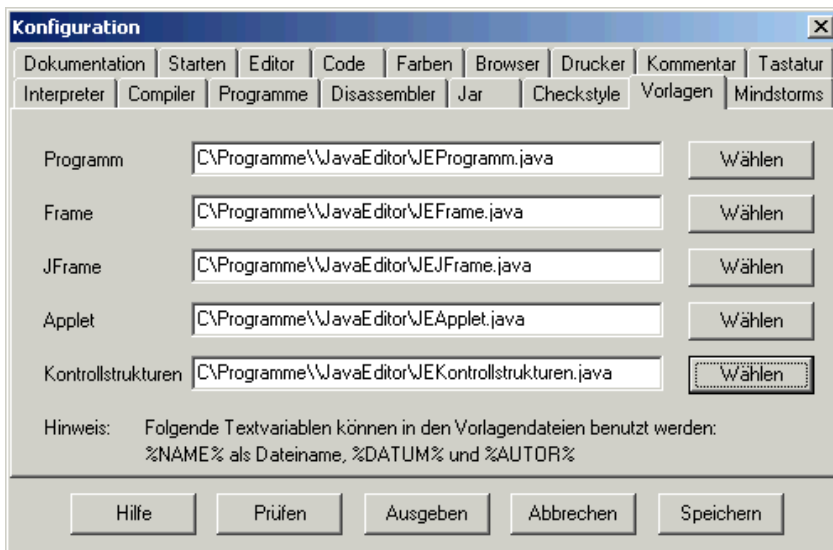


Unter *Checkstyle* ist das jar-Archiv des heruntergeladenen Checkstyle-Pakets anzugeben. Unter *Konfiguration* trägt man die xml-Datei ein, welche die Programmierrichtlinie enthält. Standardmäßig nimmt man die Programmierrichtlinien-Datei `sun_checks.xml` von Sun.

Für Unterrichtszwecke benutze ich die Konfigurationsdatei [mycheckstyle.xml \(gezippt\)](#), welche nicht ganz so streng prüft.

Vorlagen

Es gibt viele unterschiedliche Ansichten darüber, wie die Vorlage für ein Java-Programm aussehen sollte. Wenn Sie mit den vier vom Java-Editor für Konsolenprogramme, Frame- und JFrame-Programme sowie Applets benutzten Vorlagen einverstanden sind, so lassen die Vorlagen-Einstellungen frei. Ansonsten nehmen Sie eine dieser Vorlagen, modifizieren sie gemäß Ihren Vorstellungen und geben Sie auf dieser Registerkarte die modifizierte Vorlagendatei an.



In einer Vorlagendatei können Sie die Variablen `%NAME%` als Dateiname, sowie `%AUTOR%` und `%DATUM%` benutzen. Das Zeichen `|` gibt an, wo sich nach dem Laden einer Vorlage der Cursor befinden soll. Für die Funktionsfähigkeit einer Vorlage ist wichtig, dass sie die mit `//` Anfang und `//` Ende markierten Zeilen zum Einfügen von Variablen, Komponenten und Ereignisprozeduren enthält.

Auch für die Kontrollstrukturen können Sie eine Vorlagendatei benutzen. Da sich alle Vorlagen in einer Datei befinden, zeigen Schlüssel der Art `"#while"` den Anfang einer Vorlage an und das Ende der vorherigen Vorlage an. Am besten passt man die heruntergeladene Beispieldatei an seine Wünsche an.

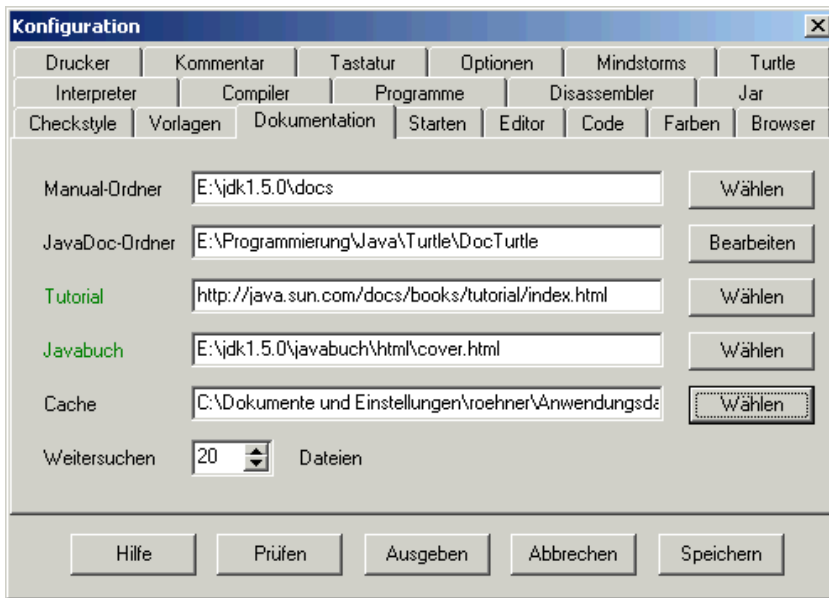
Sie können sich ein Päckchen mit fünf [Vorlagendateien](#) herunterladen.

Dokumentation

Die hier konfigurierten Pfade und Dateien werden für die Befehle im Hilfe-Menü sowie für die Code-Vervollständigung und Parameter-Hinweise benutzt.

Der Pfad für *Manual-Ordner* ist der Stammordner des [Java-Hilfesystems](#). Dieser Pfad muss gesetzt sein, damit die Code-Vervollständigung funktioniert.

Befehlsweise kann man auf <http://java.sun.com/j2se/1.5.0/docs> verweisen. Dann erfolgt der Zugriff auf die Online-Dokumentation. Zur besseren Performance werden beim Onlinezugriff die benutzten Dokumentationsdateien gecacht, also lokal abgespeichert. Standardmäßig ist als Cache-Ordner `C:\Dokumente und Einstellungen\%username%\Anwendungsdaten\JavaEditor\Cache` eingestellt.



Die Einbindung zusätzlicher Dokumentationen erfolgt über *JavaDoc-Ordner*. Hier kann man mehrere Ordner angeben, in den Dokumentationen enthalten sind, die von JavaDoc erzeugt wurden. Ist wird geprüft, ob darin eine Datei `allclasses-frame.html` enthalten ist.

Das Tutorial existiert als

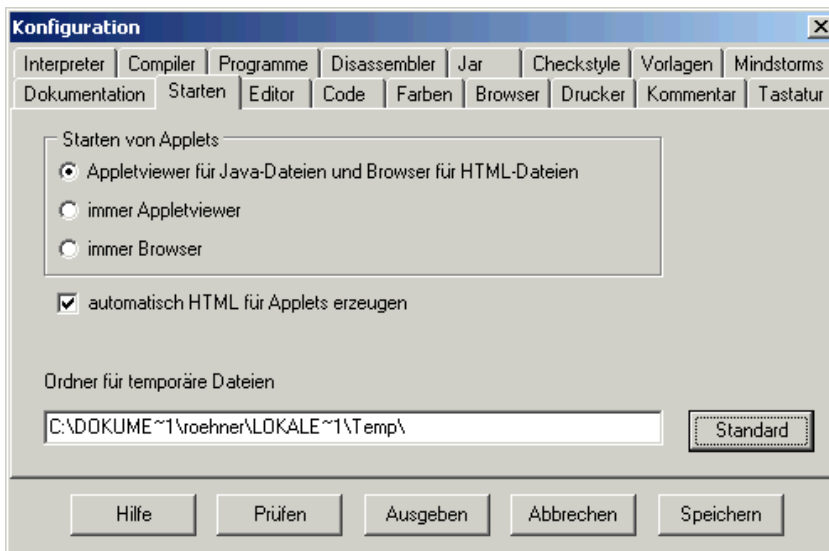
- Windows-Hilfe: >> <http://www.allimant.org/javadoc/tutorial.html>
- downloadbares HTML-Projekt: >> <http://java.sun.com/docs/books/tutorial/information/download.html>
- Online-Angebot: >> <http://java.sun.com/docs/books/tutorial/index.html>

Zusätzlich ist der >> [Download](#) und die Installation des Javabuchs in den Ordner *javabuch* sinnvoll. Gemäß der Empfehlung des Autors Guido Krügers sollten Sie das Buch in den Unterordner *html* und die Java-Beispiele in den Unterordner *examples* installieren.

Unter "Weitersuchen" stellen Sie ein, wie viele weitere Fundstellen beim Weitersuchen in der Java-Editor-Hilfe angezeigt werden.

Starten

Beim Starten einer Java-Datei mit einem Applet wird standardmäßig eine passende HTML-Datei erzeugt und dann der Appletviewer aufgerufen. Startet man hingegen eine Applet-HTML-Datei, so wird sie im Browser angezeigt. Wenn das zu undurchsichtig ist, können Sie alternativ Applets immer im Appletviewer bzw. immer im Browser starten.

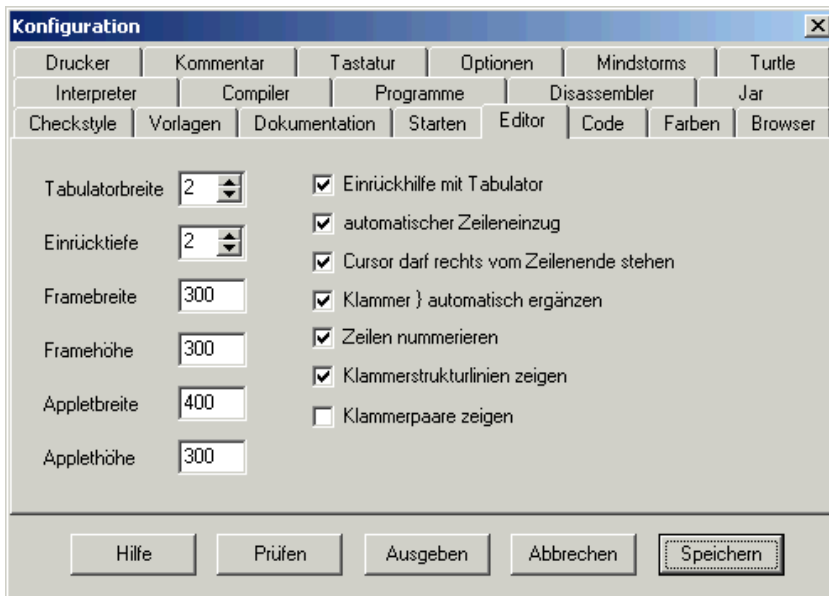


Im Ordner für temporäre Dateien legt der Java-Editor die Batchdatei `RunJava.Bat` zum Starten eines Java-Programms ab und die Fehlermeldungsdatei `error.txt`. Standardmäßig greift der Java-Editor auf den von Windows konfigurierten Ordner für temporäre Dateien zu. Beim Einsatz unter einem Terminalserver kann der Ordner für die temporären Dateien hier konfiguriert werden.

Editor

Die Tabulatorbreite läßt sich einstellen. Bei aktivierter Einrückhilfe wird statt auf die nächste Tabulatorposition auf den nächsten Wortanfang der darüberliegenden Zeile positioniert.

Die Einrücktiefe wird beim Anlegen von Programmen über die Symbole der Programm-Komponenten, beim Einfügen von AWT-Komponenten und Kontrollstrukturen benutzt.



Der automatische Zeileneinzug zieht bei Eingabe von Return die nächste Zeile auf das erste nicht leere Zeichen der vorherigen Zeile ein.

Die standardmäßige Anzeige der Zeilennummerierung und die Standardvorgaben für Frame- bzw. Appletgröße lassen sich festlegen.

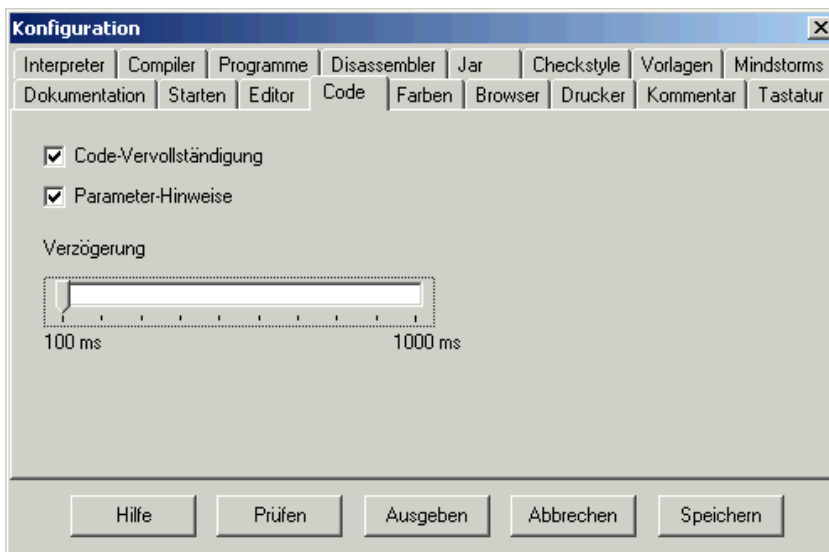
Bei Eingabe einer öffnenden { -Klammer erzeugt der Editor bei dieser Option eine zugehörige } -Klammer. Wer das nicht mag, kann es hier Abschalten.

Zeilennummerieren schaltet automatisch die Zeilennummerierung beim Öffnen von Dateien ein, analoges gilt für die Klammerstrukturlinien und Klammernpaare.

Mit den Angaben *Framebreite* und *Framehöhe* können Sie Standardvorgaben für die Größe des Anwendungsfensters machen. Diese Vorgaben werden benutzt, wenn die Option *Framewerte statt pack()* aktiviert ist. Ist sie nicht aktiviert, so wird *pack()* in die Programmschablone geschrieben, womit die Größe des Programmfensters in Abhängigkeit von Größe und Layout der darin enthaltenen Komponenten ermittelt wird.

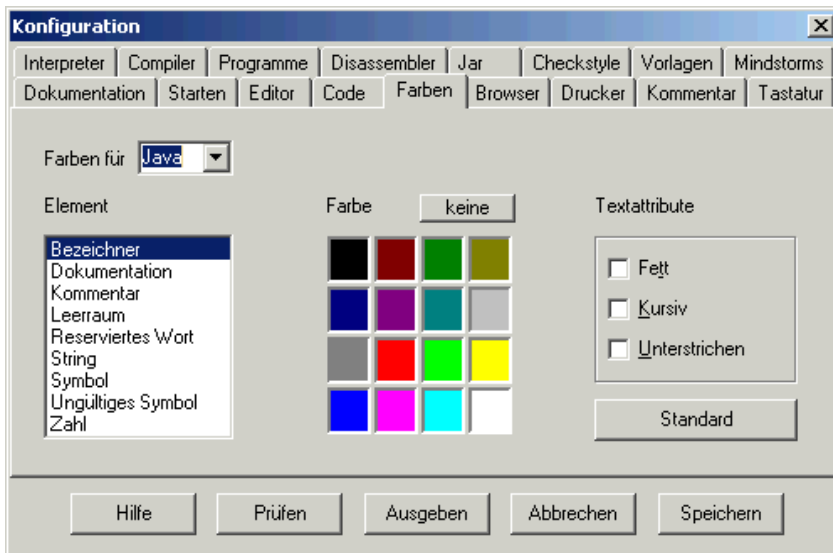
Code

Auf dieser Registerkarte aktivieren Sie die Code-Vervollständigung und Parameter-Hinweise und legen deren Verzögerung in Millisekunden fest.



Farben

Die Farben für die Syntaxhervorhebung von Java- und HTML-Dokumenten sowie für die Klammerstrukturlinie und Klammernpaare lassen sich konfigurieren.

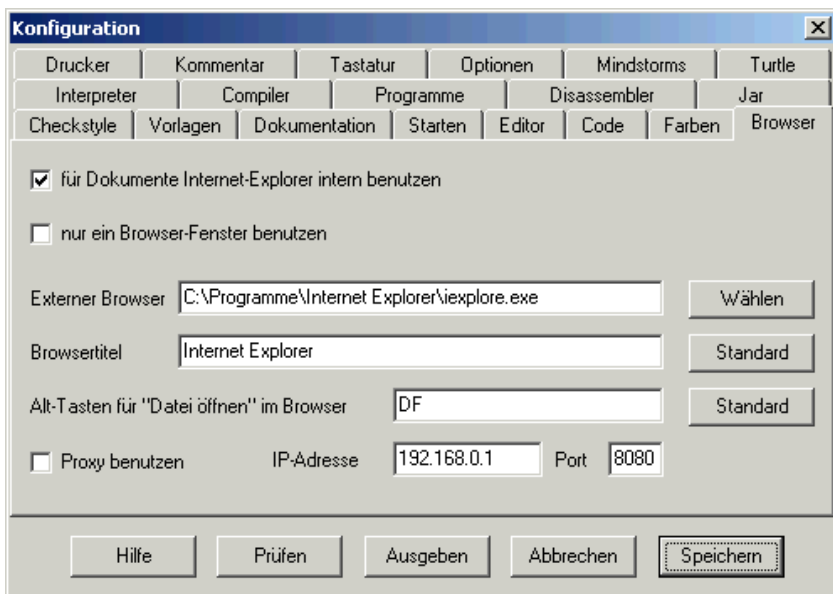


Wählt man links ein Syntax-Element aus, so werden dessen Vorder- und Hintergrundfarben und Textattribute rechts angezeigt. Anschließend stellen Sie für dieses Element die gewünschte Farben und die Textattribute ein. Der Schalter *Standard* stellt für alle Syntax-Elemente die Standard-Farben ein. Mit *keine* werden Vorder- und Hintergrundfarbe zurückgesetzt.

Browser

Ist der Internet-Explorer ab Version 4 installiert, so können Sie ihn intern im Java-Editor zur Anzeige der Dokumentation oder auch zum Ausführen von Applets benutzen. Der interne Modus steht nur für den Internet-Explorer aber nicht für andere Browser wie z. B. Navigator oder Opera zur Verfügung. Sie können also zum Beispiel den Internet-Explorer zur internen Anzeige der Dokumentation benutzen und für das Testen von Applets den Netscape Navigator als externen Browser.

Aktivieren Sie "nur ein Browser-Fenster benutzen", wenn insgesamt nur ein einziges internes Browser-Fenster benutzt werden soll.

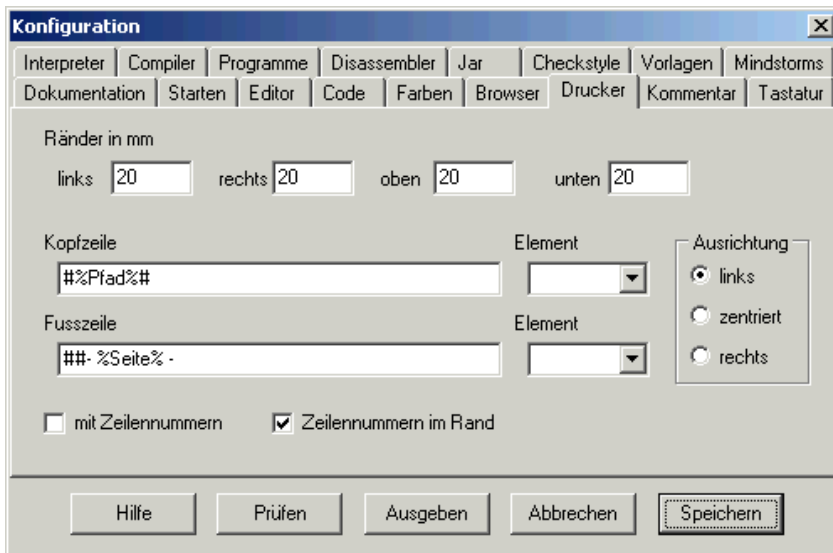


Soll ein Browser extern benutzt werden, so wählen Sie unter *externer Browser* den gewünschten Browser aus. Zusätzlich sind für den externen Browser die Angaben *Browsertitel* und *Alt-Tasten für Datei öffnen im Browser* nötig. Der Java-Editor prüft zunächst, ob der externe Browser schon geladen ist. Dazu benötigt er den Browsertitel, also zum Beispiel *Netscape Navigator*, *Internet Explorer* oder *Opera*. Ist er schon geladen, so wird die darzustellende HTML-Seite über das Datei-Menü des Browsers mit dem Befehl *Öffnen* in das sich öffnende Adressfeld eingetragen. Dazu wird das Datei-Menü mit Alt-D aktiviert und mit F der Öffnen-Befehl ausgewählt. Daraus resultiert die Angabe DF im Konfigurationsdialog. In einer englischsprachigen Version des Browsers muss der Eintrag entsprechend angepasst werden.

Aktivieren Sie falls nötig *Proxy benutzen* und geben Sie dann die IP-Adresse und den Port des Proxys an. Ein aktivierter Proxy wird für die Update-Funktion und die Code-Vervollständigung beim http:-Zugriff auf die Dokumentation benutzt.

Drucker

Auf der Registerkarte *Drucker* legen Sie die Druckränder fest, definieren die Kopf- und Fußzeile und bestimmen die Verwendung von Zeilennummern.

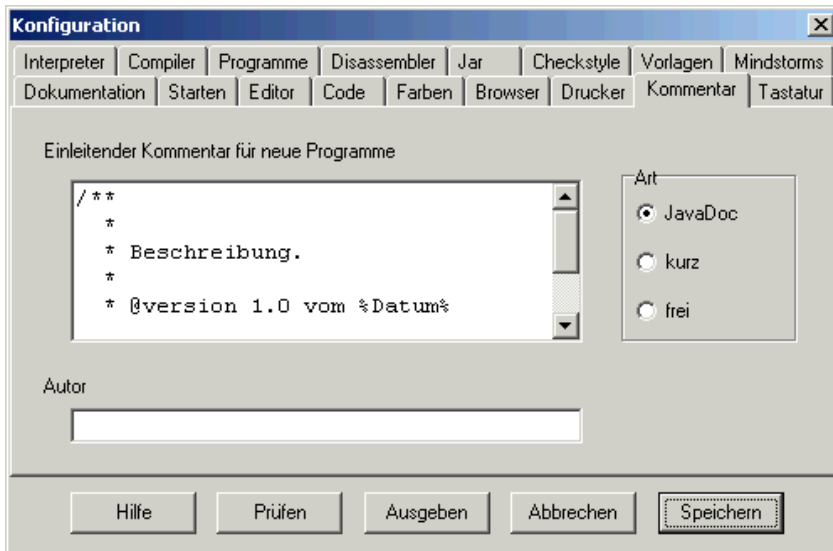


Kopf- und Fußzeile sind in die drei Bereiche Links, Mitte und Rechts gegliedert, welche linksbündig, zentriert bzw. rechtsbündig gedruckt werden. Die Trennung der Bereiche wird durch die #-Zeichen festgelegt. In der Kopf- und Fußzeile können Sie neben feststehendem Text noch folgende variable Elemente drucken:

%Datei%	der Dateiname
%Pfad%	der Dateiname inklusive Verzeichnispfad
%Datum%	das Druckdatum
%Zeit%	die Druckzeit
%Seite%	die Nummer der Druckseite
%Seitenzahl%	die Gesamtzahl der Druckseiten

Kommentar

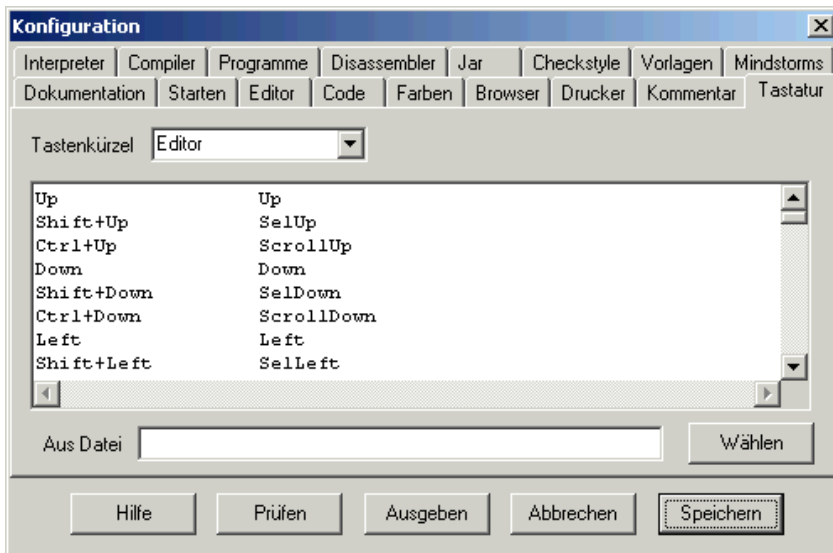
Jedes neue Java-Programm wird automatisch mit einem einleitenden Kommentar versehen.



Zur Auswahl steht ein Kommentar in JavaDoc-Art, ein kurzer Kommentar aus Autor und Datum, sowie ein frei gestalteter Kommentar. Als variable Elemente können Sie im Kommentar %Autor% und %Datum% benutzen. Die %Autor%-Angabe wird dem Feld *Autor* entnommen, %Datum% entspricht dem aktuellen Datum der Programmerzeugung.

Tastatur

Die Registerkarte *Tastatur* zeigt für den Editor und das Menü-System die Tastenbelegung an.



Zusätzlich können in zwei Dateien Tastenkürzel definiert werden.

1. In der Datei *JETastatur.txt* im Systemverzeichnis kann der Administrator Tastenkürzel vorgeben.
2. Benutzer können sich in einer weiteren Datei zusätzliche Tastenkürzel definieren.

Eine Tastenkürzel-Definitionsdatei ist eine Textdatei. Die Definition eines Tastenkürzels beginnt mit *Tastenkürzel:<Tastenbezeichnung>*

Tastenbezeichnungen werden wie folgt geschrieben:

```
F1, ..., F12
Strg+A, ..., Strg+Z, Strg+F1, ..., Strg+F12
Umsch+Strg+A, ..., Umsch+Strg+Z, Umsch+Strg+F1, ..., Umsch+Strg+F12
Alt+A, ..., Alt+Z, Alt+F1, ..., Alt+F12
Umsch+Alt+A, ..., Umsch+Alt+Z, Umsch+Alt+F1, Umsch+Alt+F12
Strg+Alt+A, ..., Strg+Alt+Z, Strg+Alt+F1, ..., Strg+Alt+F12
```

Innerhalb der Tastenkürzel-Definition kann mit | die Position des Cursors angegeben werden. Eine Kürzel-Definition muss mit *Tastenkürzel:Ende* beendet werden.

Beispiel

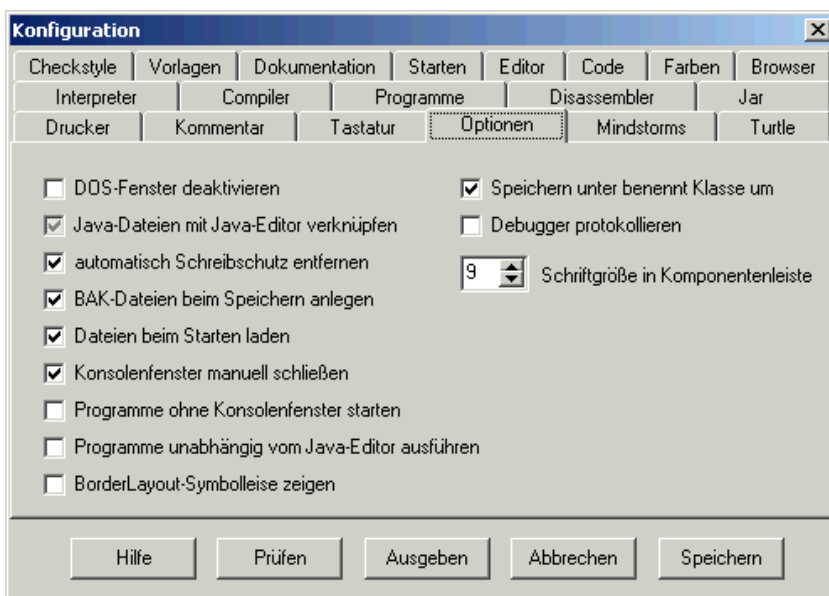
```
Tastenkürzel:Strg+I
import |
Tastenkürzel:Ende
```

belegt Strg+I mit dem Text *import* wobei der Cursor ein Leerzeichen hinter import positioniert wird.

Eine Beispieldatei wird bei der Installation erstellt.

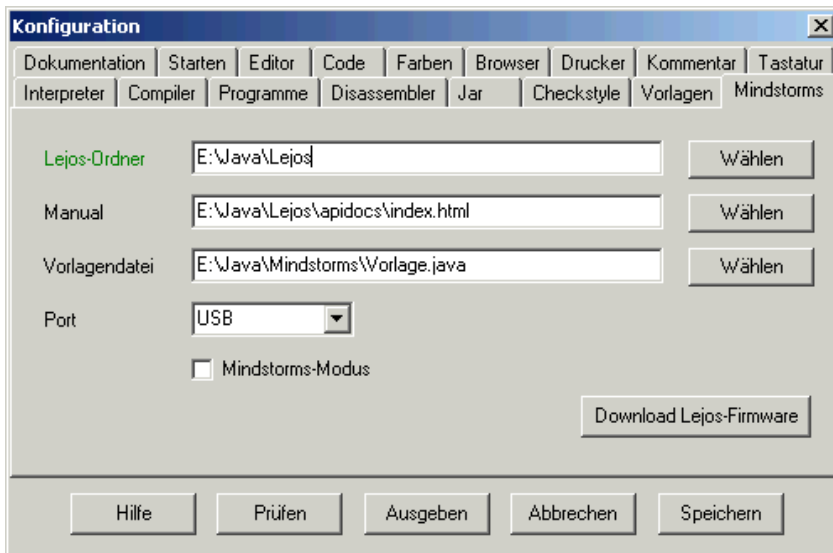
Optionen

Auf dieser Registerkarte können Sie verschiedene Einstellungen vornehmen.



Mindstorms

Die Lego-Mindstorms-Modelle können mit Java-Programmen programmiert werden. Dazu muss man das » [Lejos-System](#) installieren und auf den RCX-Baustein die Lejos-Firmware aufspielen.



Unter *Lejos-Verzeichnis* geben Sie das Lejos-Installationsverzeichnis an. Es empfiehlt sich, auch die [Lejos-Dokumentation](#) zu installieren, um mit der Lejos-API arbeiten zu können.

Sie können eine spezielle Vorlagen-Datei für Mindstorms-Programme anlegen und hier eintragen. Für diese Vorlagen-Datei gelten die gleichen Variablen wie bei den anderen Vorlagen-Dateien.

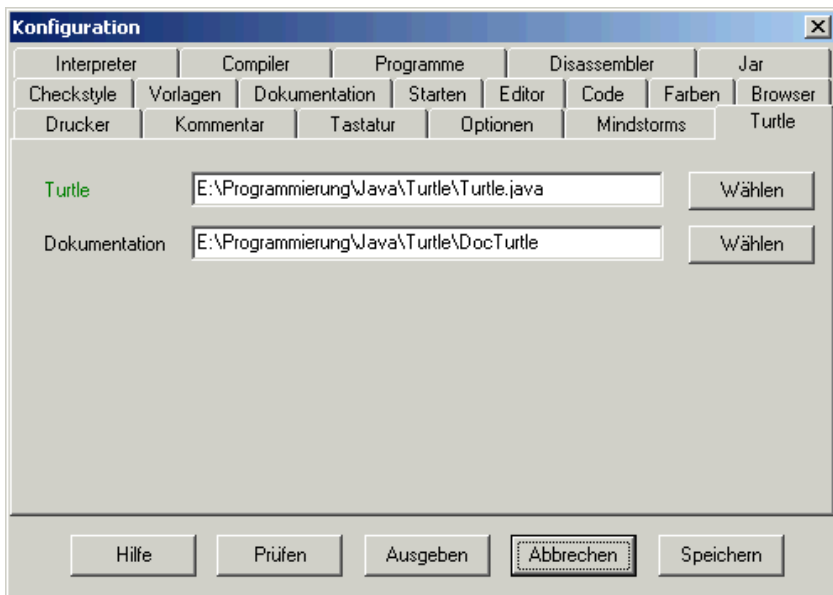
Unter Port ist der Port für den Infrarot-Transmitter anzugeben, beim RCX 2.0 also USB.

Mit dem Schalter *Mindstorms-Modus* schalten Sie den Java-Editor um. Das Programm-Symbol der Komponentenleiste erzeugt dann den Rumpf eines Java-Programmes für Mindstorms. Der Compiler compiliert speziell für den RCX und das Starten-Symbol überträgt das compilierte Programm auf den RCX-Baustein.

Unterrichtliche Beispiele finden Sie auf [lehrer-online](#).

Turtle

Für Unterrichtszwecke steht eine Turtle zur Verfügung. Sie können diese [herunterladen](#) und dann auf dieser Registerkarte konfigurieren.



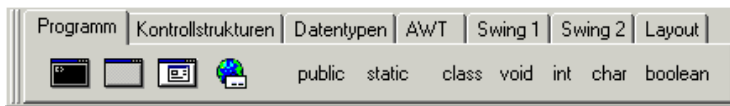
Hilfe-Menü

Das Hilfe-Menü bietet den direkten Zugang zur Dokumentation des Java-SDK. Die einzelnen Befehle rufen standardmäßig im SDK-Verzeichnis folgende Seiten bzw. Dateien auf:

Index	JDK-Hilfe-Fenster
API	\docs\api\index.html
Demos	\docs\relnotes\demos.html
Tutorial	\tutorial\tutorial.hlp
Javabuch	\javabuch\html\cover.html

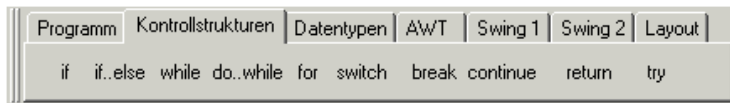
Komponenten

Mit der Programm-Komponentenleiste legen Sie Konsolen-Programme, AWT- und Swing-Anwendungen bzw. Applets an.



Die vorgegebenen Vorlagen können Sie über die Registerkarte Vorlagen durch eigene ersetzen.

Kontrollstrukturen und Datentypen werden an der Cursorposition in den Quelltext eingefügt.



AWT- und Swing-Komponenten werden unterstützt. Sie können aus den Registern AWT, Swing1 und Swing2 abgerufen werden.



Die Platzierung einer Komponente für das BorderLayout lässt sich über die zugehörige BorderLayout-Symboleiste vorwählen.



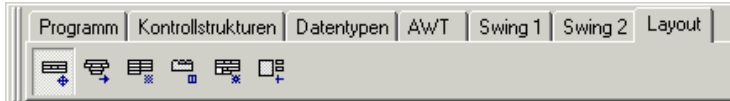
In diesem Fall ist die mittlere Schaltfläche ausgewählt, weswegen als Platzierung *Center* benutzt wird. Sie können die Platzierung deaktivieren, in dem Sie die aktivierte Schaltfläche der BorderLayout-Symboleiste noch mal anklicken.

Die Komponenten werden direkt vor den Kommentaren

```
// Ende Variablen
// Ende Komponenten
// Ende Ereignisprozeduren
```

eingefügt, auf die Cursor-Position wird dabei keine Rücksicht genommen. Für jede Komponente wird eine Variable deklariert, im Komponentenbereich des Konstruktors wird sie erzeugt, bei Buttons wird automatisch eine Ereignisprozedur angelegt.

Mit der Layout-Komponentenleiste wählen Sie unterschiedliche Layouts aus:



Editor

Der Editor basiert auf der » [SynEdit-Komponente](#). Neben den üblichen Editor-Funktionen bietet er Ein- und Ausrücken von Blöcken (Umschalt+Strg+I [indent] bzw. Umschalt-Strg-U [unindent]), Zeile löschen (Strg+Y), bis zum Ende der Zeile löschen (Umschalt+Strg+Y) sowie die Verwaltung von zehn Lesezeichen (setzen/löschen Umschalt+Strg+#, anspringen Strg+#). Die Lesezeichen können auch direkt in der Lesezeichenspalte bzw. mit dem Lesezeichen-Symbol gesetzt und gelöscht werden.

Zukunft

Gemäß den unterrichtlichen Anforderungen und der verfügbaren Zeit wird der Java-Editor weiterentwickelt.

Gerhard Röhner·26.08.2006
 Bildungsserver Hessen »
[http://lernen.bildung.hessen.de/informatik/javaeditor/index.htm/index_ ...](http://lernen.bildung.hessen.de/informatik/javaeditor/index.htm/index_...)
 ... html