Java-Tutorium WS 09/10

Tutorial: Eclipse Debugger

Java-Tutorium WS 09/10
Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

Was ist der Eclipse Debugger?

Die Eclipse Plattform stellt einige sehr hilfreiche Features zum Programmieren bereit. Eines dieser Features ist der *Debugger.*

Mithilfe des Debuggers lässt sich eine Java-Applikation zur Laufzeit analysieren.

Die Debug Perspektive (oder *Debug perspective*) bietet die Möglichkeit die einzelnen Programmschritte (Objekterzeugung, Methodenaufrufe, Variablenzuweisungen etc.) abzulaufen um das Programm auf Fehler zu überprüfen.

Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

Breakpoints

Ein Breakpoint unterbricht die Ausführung des Programms an der Position an dem der Breakpoint gesetzt wurde.

Breakpoints können aktiviert und deaktiviert werden via Kontextmenü im Breakpoint View.

• Wenn ein aktivierter Breakpoint erreicht wird unterbricht dieser den momentanen Thread (*Prozess*). Aktive Breakpoints sind mit einem blauen Kreis gezeichnet [Breakpoint].

• Ein deaktivierter Breakpoint suspendiert nicht den momentanen Thread. Dektivierte Breakpoints sind mit einem weißen Kreis gezeichnet [Disabled Breakpoint].

Breakpoints werden vertikal im Java-Editor (Seitenrand) und im Breakpoints View angezeigt.

Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

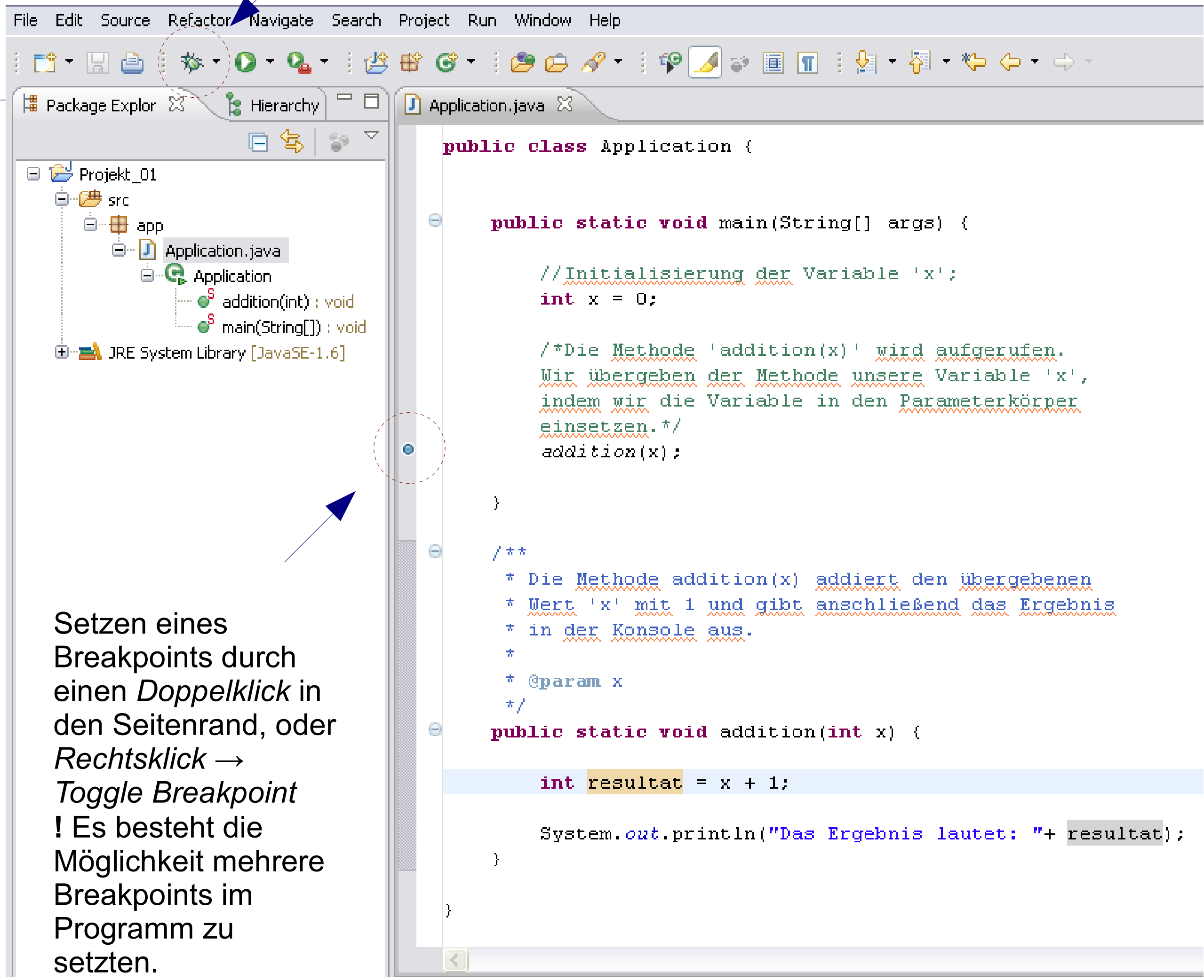
Den Debugger starten

Um den Debugger zu starten und in die Debug-Perspektive zu wechseln gibt es mehrere Möglichkeiten...

• *Rechtsklick auf die Main-Klasse* im Package Explorer → *Debug As* → *Java Application* oder...

• *Debug Application* (Käfer) → *Debug As* → *Java Application*

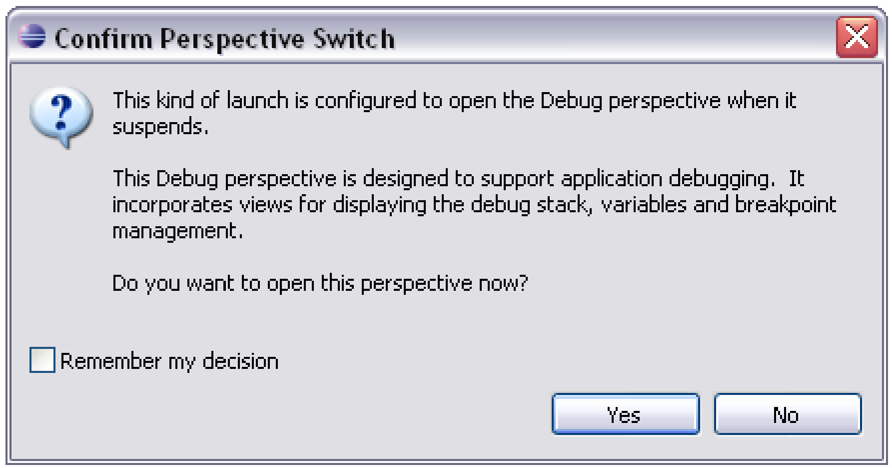
Debug Application



Setzen eines Breakpoints durch einen *Doppelklick* in den Seitenrand, oder *Rechtsklick* → *Toggle Breakpoint*
! Es besteht die Möglichkeit mehrere Breakpoints im Programm zu setzen.

Java-Tutorium WS 09/10
Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

Confirm Perspective Switch



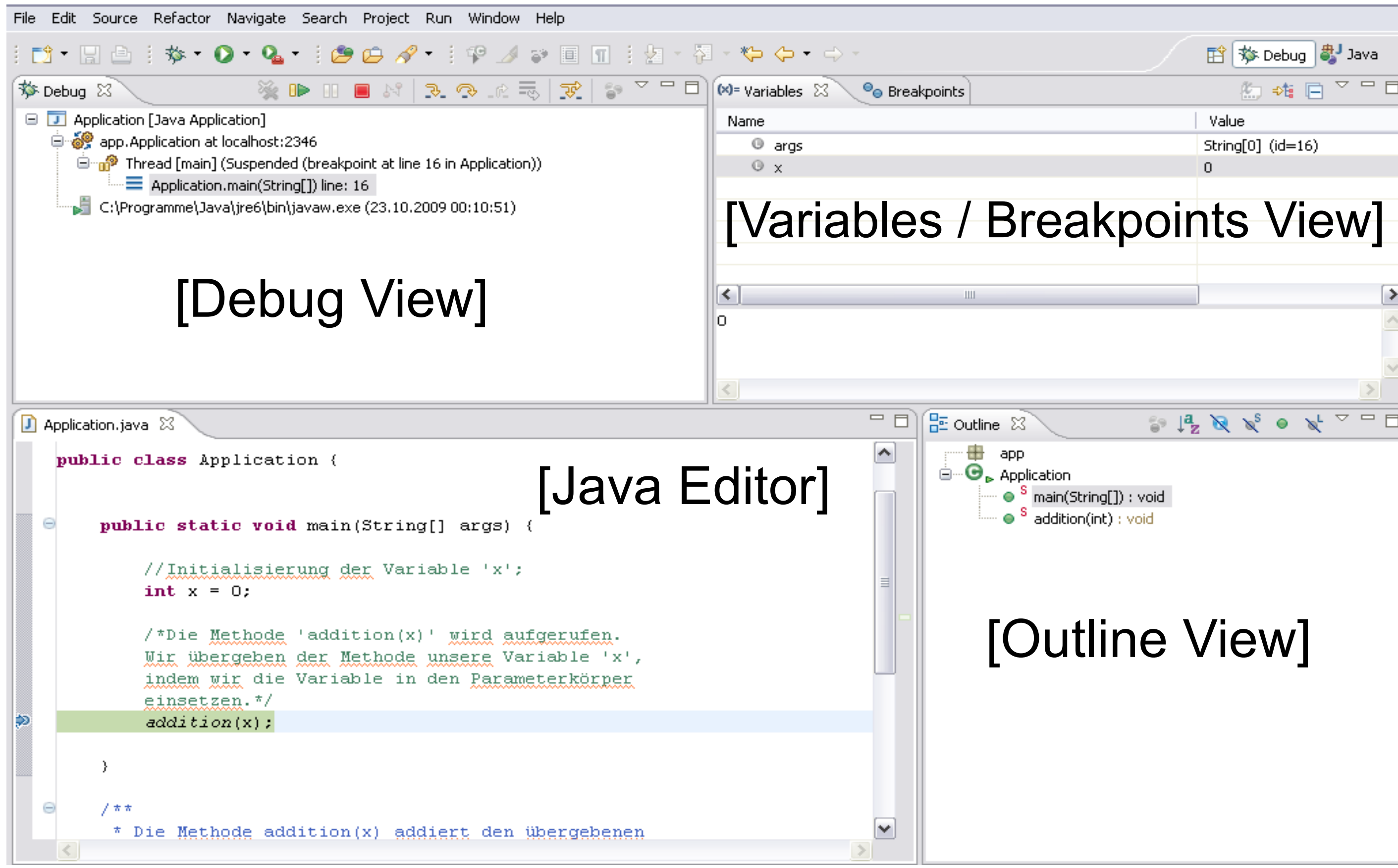
- Es erscheint ein Dialog, welches den Benutzer fragt, ob Eclipse die *Debug-Perspektive* öffnen soll → yes

Java-Tutorium WS 09/10
Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

Debug Perspektive

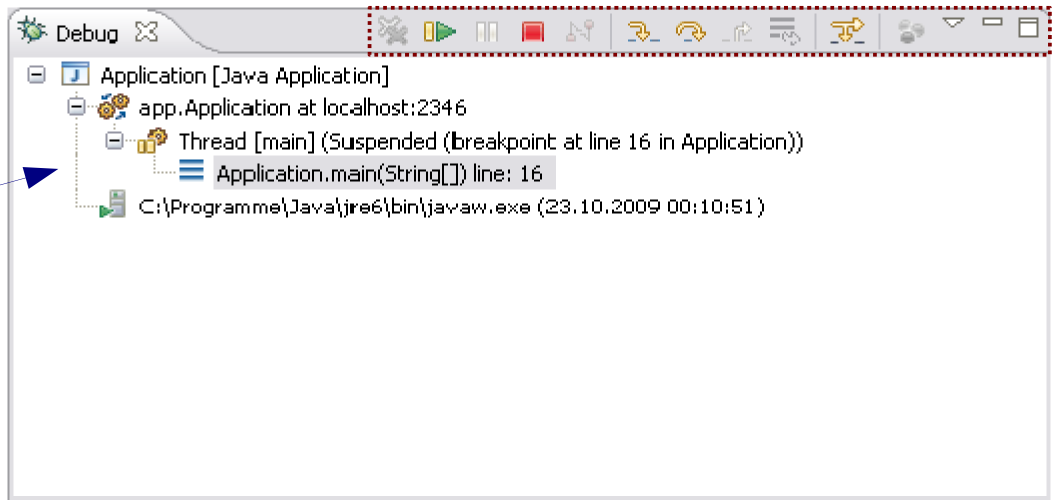
Eine neue *Perspektive* wird in Eclipse geöffnet (*Debug Perspective*)

- Debug View
- Variables View
- Breakpoints View
- Java Editor
- Outline View



Debug View

Execution Stack



• Die *Debug View Toolbar* bietet einige Funktionen (*Resume*, Suspend, *Step Into, Step Over* etc.) um den Debugger zu navigieren

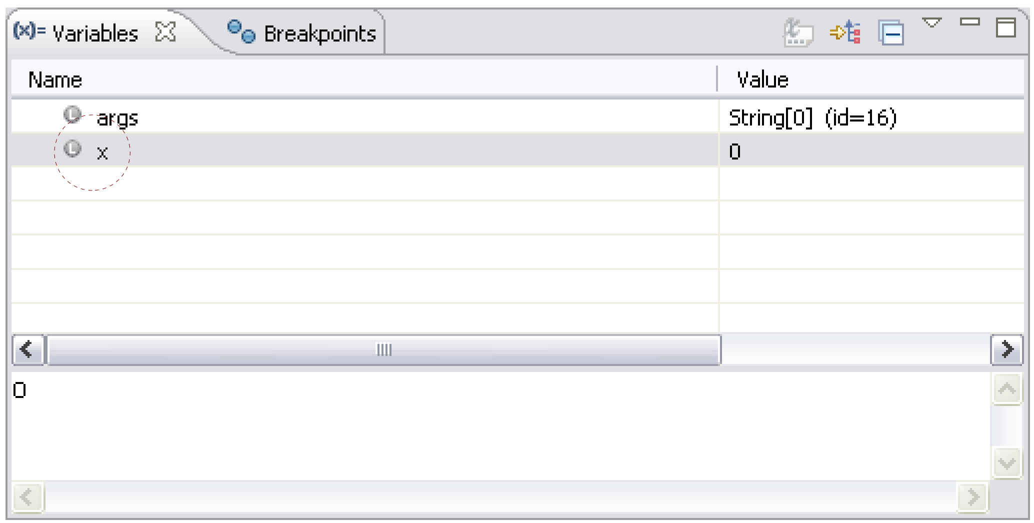
• Im Debug View (Fenster) lassen sich die unterbrochenen Threads / Prozesse beobachten

• Jeder Thread im Programm erscheint als ein Knoten im Baum

Debug View Toolbar

	Relaunch	This command re-launches the selected debug target.	Context menu
	Resume	Resumes a suspended thread.	Context menu, Run menu and view action
	Step Into	Steps into the highlighted statement.	Context menu, Run menu and view action
	Step Over	Steps over the highlighted statement. Execution will continue at the next line either in the same method or (if you are at the end of a method) it will continue in the method from which the current method was called. The cursor jumps to the declaration of the method and selects this line.	Context menu, Run menu and view action
	Step Return	Steps out of the current method. This option stops execution after exiting the current method.	Context menu, Run menu and view action
	Suspend	Suspends the selected thread of a target so that you can browse or modify code, inspect data, step, and so on.	Context menu, Run menu and view action
	Terminate	Terminates the selected debug target.	Context menu, Run menu and view action
	Terminate & Relaunch	Terminates the selected debug target and relaunches it.	Context menu
	Terminate All	Terminates all active launches in the view.	Context menu
	Use Step Filters	Toggles step filters on/off. When on, all step functions apply step filters.	Context menu, Run menu and view action

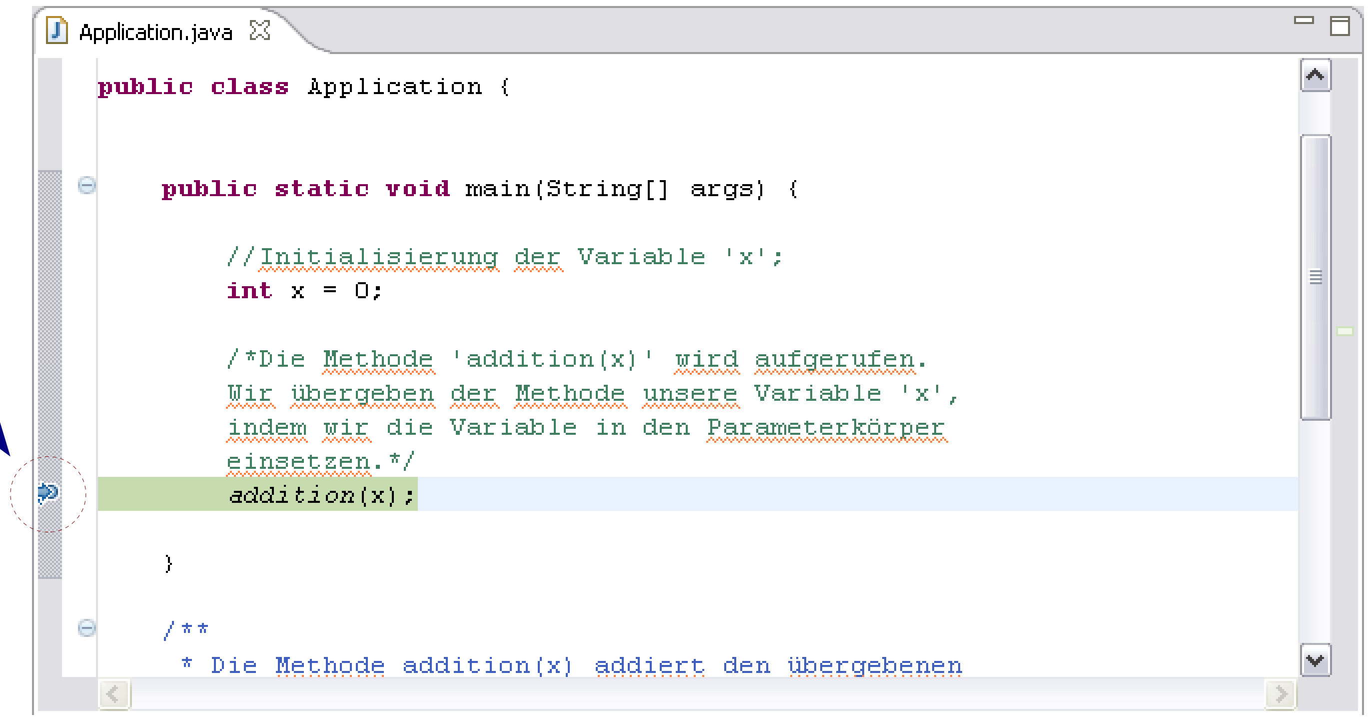
Variables / Breakpoints View



• In diesem *View* lassen sich Variablen/Objekte und deren Werte beobachten

Der Java Editor innerhalb der Debug Perspektive

- Das Programm wird am ersten *Breakpoint* im Programm unterbrochen

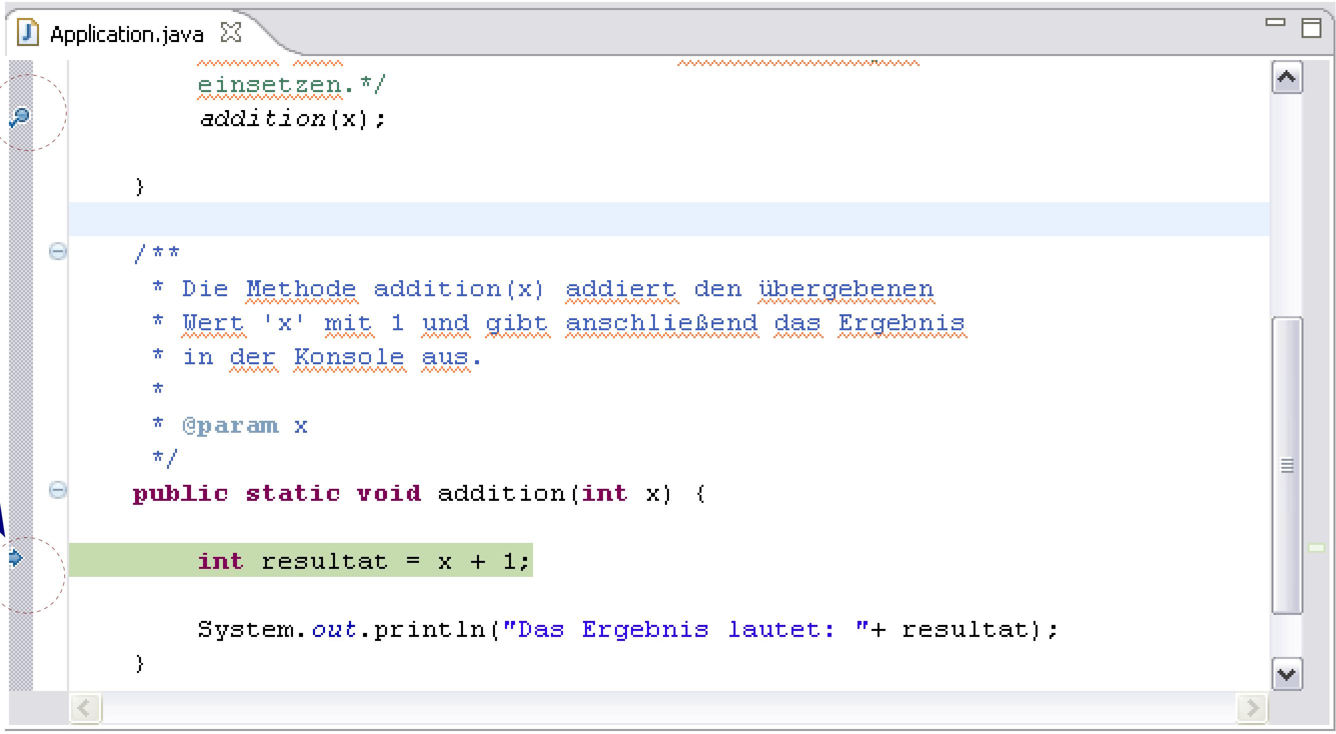


Java-Tutorium WS 09/10
Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

Der Java Editor innerhalb der Debug Perspektive 1

- Durch den Button *Step Into,* der Debug Toolbar, springt der Debugger in die Methode addition(x)

- Die im Augenblick auszuführende Codezeile ist mit einem grünen Balken im Hintergrund markiert

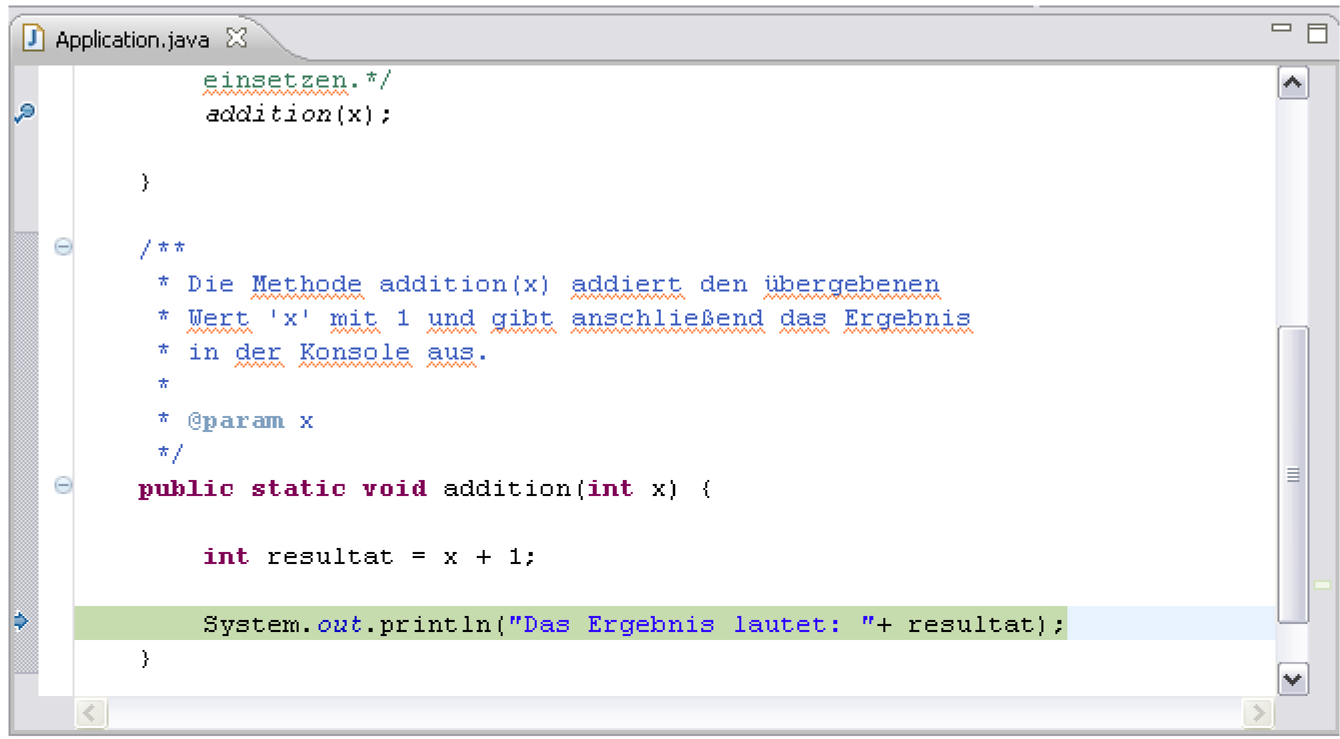


Java-Tutorium WS 09/10
Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

Der Java Editor innerhalb der Debug Perspektive 2

- Durch weiteres klicken des Buttons *Step Into* werden die nächsten Codezeilen ausgeführt

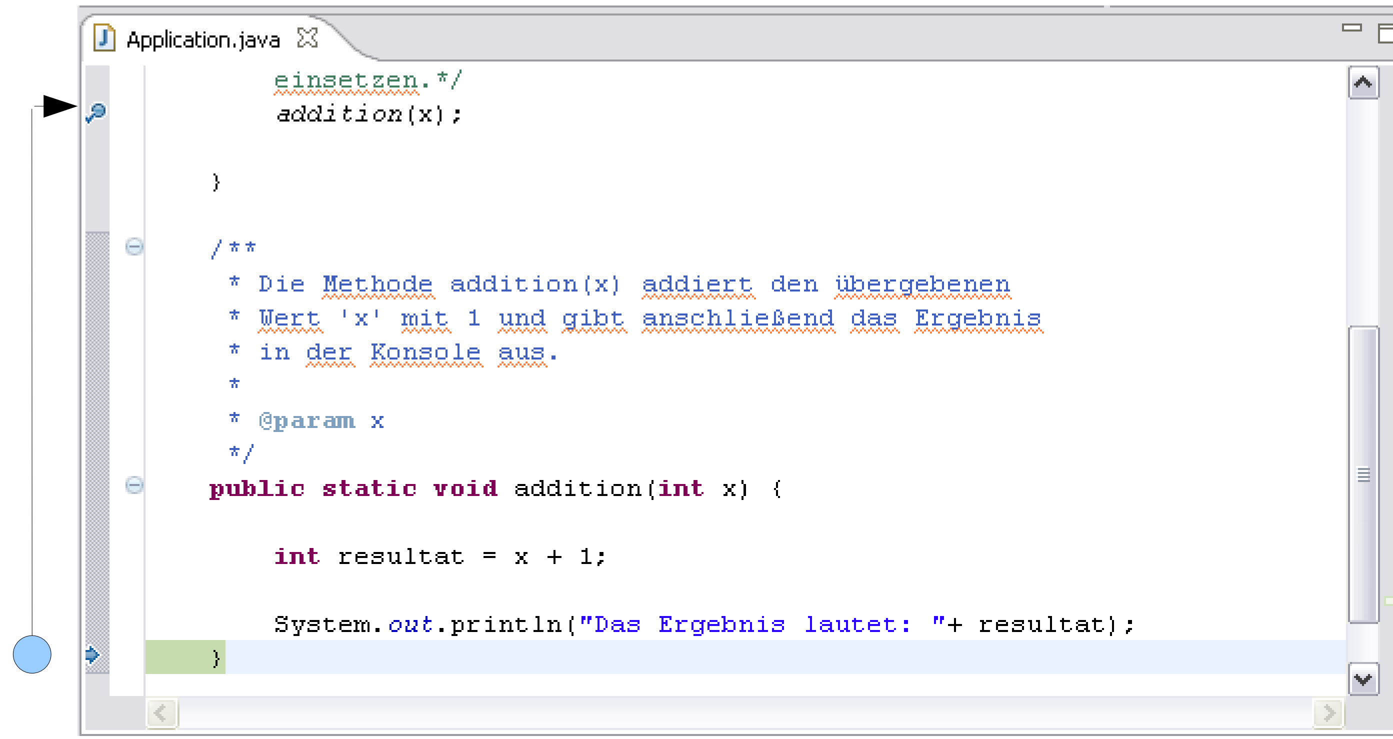
- Man darf an dieser Stelle nicht die anderen Views vergessen, wie z.B. den Variables View, der nach jeder Anweisung die neuen bzw. veränderten Werte der Variablen des Programms anzeigt



Java-Tutorium WS 09/10
Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

Der Java Editor innerhalb der Debug Perspektive 3

- Wenn der Debugger am Ende eines Methodenkörpers angekommen ist, springt er an den vorherigen Breakpoint zurück bzw. in die Zeile die die Methode aufgerufen hat



Hotswap Bug Fixing: On-the-fly code fixing

• Seit der Java Virtual Machine (JVM) V.1.4 unterstützt Eclipse ein weiteres Feature namens *Hotswap Bug Fixing*

• Das Feature erlaubt dem Programmieren den Code während der *Debug Session* zu manipulieren

• Um diese Funktion nutzen zu können, muss man lediglich die entsprechenden Veränderungen im Java Editor vornehmen und *Resume* innerhalb der *Debug Toolbar* ausführen

• Ein häufiger Fehler ist der Aufruf von *Debug* oder *Run* anstelle von *Resume*, während man sich in einer Debug Session befindet

• Der Aufruf von *Debug* oder *Run* ruft eine weitere Debug Session auf anstatt die aktuelle fortzuführen

• Falls das Programm nicht vollständig ausgeführt wurde, obwohl Du mit dem Debuggen fertig bist, führe *Terminate* im Kontextmenü aus

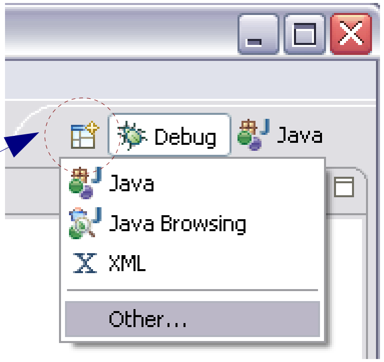
Umschalten zwischen der Java und Debug Perspektive

Die beiden Buttons (*Debug* / *Java*) ermöglichen das hin und her schalten zwischen der Java und der Debug Perspektive

Der Button mit dem hellen Hintergrund zeigt die momentan ausgewählte Perspektive an

Tutor: Mihail Atanassov
Email: tutoren@spinfo.uni-koeln.de
Sprachliche Informationsverarbeitung
Universität zu Köln

...und noch mehr Perspektiven



• Durch anklicken dieses Symbols (*Open Perspective*) erscheint ein Drop-Down Menü mit weiteren Perspektiven

• Bspw. gibt es, wie hier zu sehen, speziell eine für XML konzipierte Perspektive...

Quellen

- http://help.eclipse.org/help32/index.jsp?topic=/org.eclipse.jdt.doc.user/reference/
- http://www.ibm.com/developerworks/library/os-ecbug/