

# Installation und Test von *Eclipse*



Autor: Thomas Letsch

**Letsch Informatik**

*Beratung, Ausbildung und Realisierung in Software-Engineering*

Binzholzweg 17  
8636 Wald

<http://www.letsinfo.ch>

Release: 9.0  
Date: 21.10.2010

Copyright © 2003-2010 by Letsch Informatik @ CH-8636 Wald

## Inhaltsverzeichnis

1	Einführung .....	2
2	Installation.....	3
3	Verifikation .....	4
4	Debugging.....	9

### 1 Einführung

*Eclipse* ist eine Open-Source-Projekt, dass ein Applikations-Framework speziell für Entwicklungs-Umgebungen zur Verfügung stellt.

Mittels eines definierten *Plug-In*-Mechanismus können Erweiterungen von Dritthersteller integriert werden.

Die Grundversion enthält als Erweiterung bereits eine *Java-Entwicklungsumgebung*.

URL: <http://www.eclipse.org>

Die nachfolgende Beschreibung zeigt die *Installation* und *Verifikation* von *Eclipse* Version 3.5.2 (Build id: M20100211-1343).

## 2 Installation

Die Installation von *Eclipse* setzt eine bereits vorhandene Java-Umgebung voraus, z.B. *JavaTM 2 Platform, Standard Edition* ( <http://java.sun.com/j2se> ).

Im Falle von *J2SE6.0* für MS Windows z.B. mit *jdk-6u20-windows-i586.exe* oder *jdk-6u20-windows-x64.exe* .

Ein *Eclipse-Release* kann von <http://www.eclipse.org/downloads> runtergeladen werden.

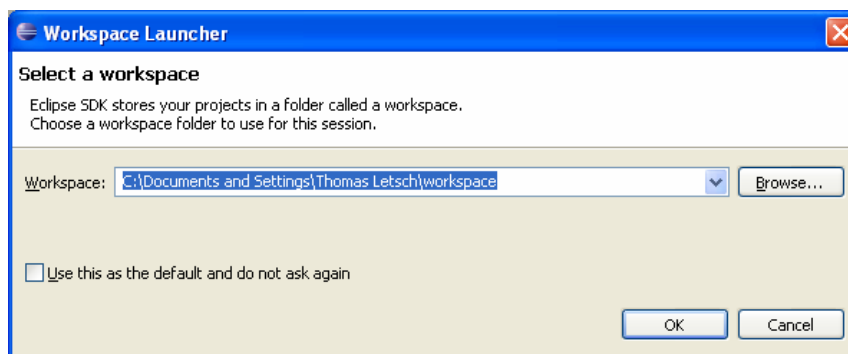
Für den Fall der Version 3.5.2 für MS Windows erhält man die ZIP-Files:

***eclipse-SDK-3.5.2-win32.zip*** oder  
***eclipse-SDK-3.5.2-win32-x86\_64.zip***.

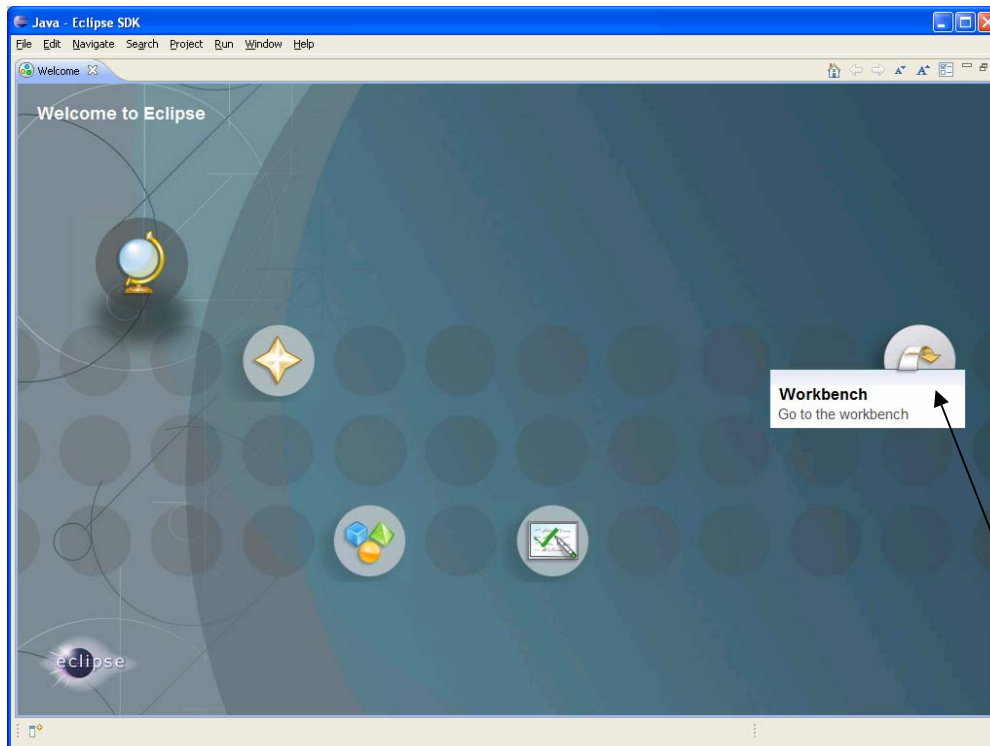
Installation (für MS Windows 32-bit):

- ***eclipse-SDK-3.5.2-win32.zip*** in beliebiges Ziel-Directory entpacken, z.B. ***C:\Program Files\***.  
Grösse auf Disk: ~ 180 MB
- Applikation starten (gemäss obigem Installationspfad):  
***C:\Program Files\eclipse\eclipse.exe***  
Es folgt eine automatische Installation von Eclipse.  
Bemerkung: es werden keine *Registry*-Einträge erzeugt ! ;-)

Darauf folgt der ***Workspace Launcher***-Dialog. Mit diesem kann festgelegt werden, wo Eclipse seine Meta-Daten ablegen soll (das entspricht den Daten, welche sonst typischerweise in der *Registry* abgelegt werden).

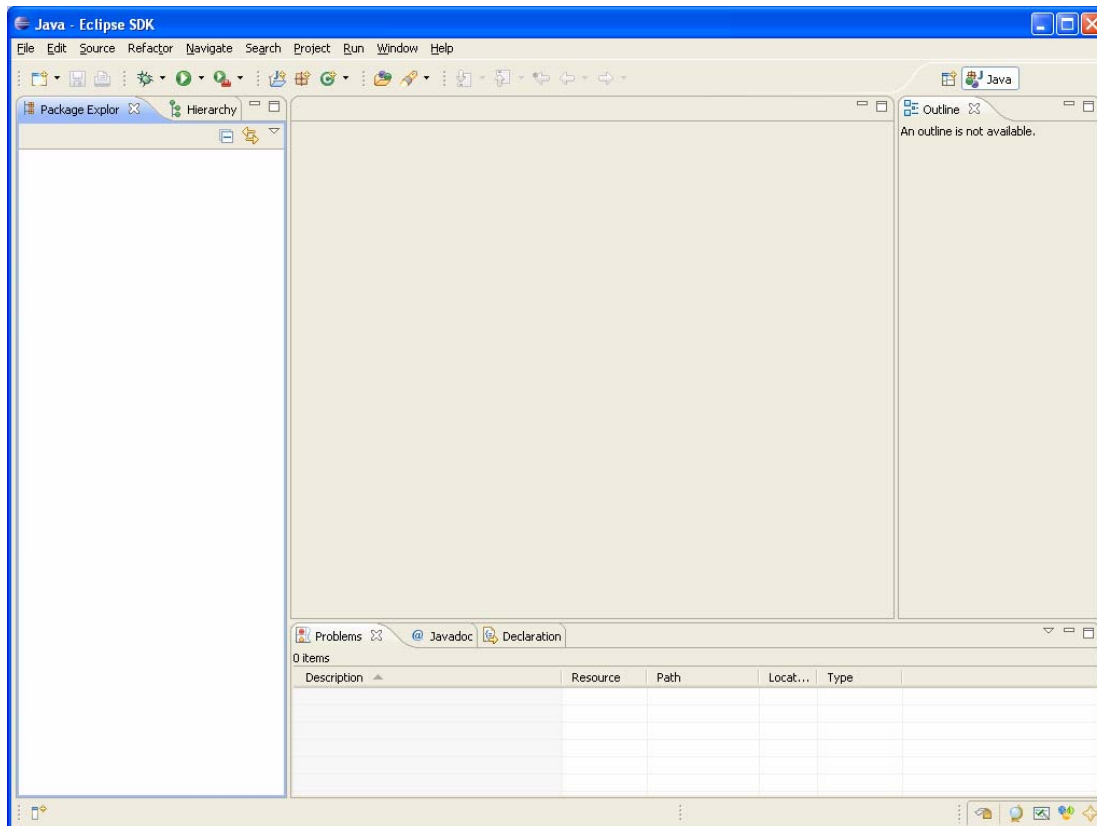


Daraufhin erscheint das *Eclipse*-Window.



Eclipse enthält für die verschiedenen Anwendungsbereiche mehrere sog. *Perspektiven*.  
Um die Perspektive für die Java-Programmierung zu öffnen: Das Icon *Workbench* drücken

Die *Java-Perspektive* wird geöffnet:

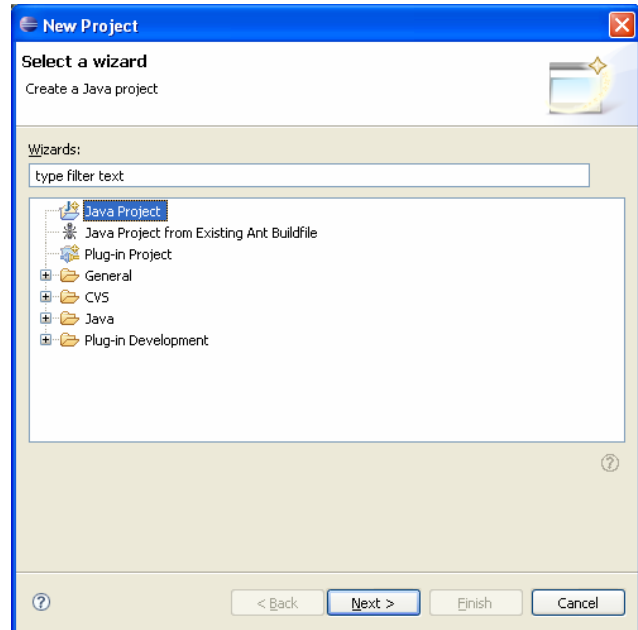


### 3 Verifikation

Zur Verifikation ein kleines Java-Projekt aufsetzen mit *InstallTestJava.zip* :

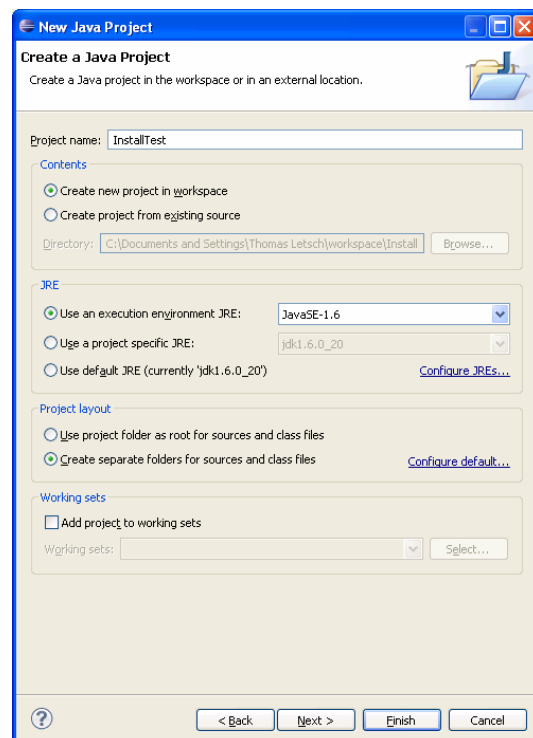
Ein neues Projekt in Eclipse anlegen mit *Menü:File>New>Project...*

*Java Project* selektieren



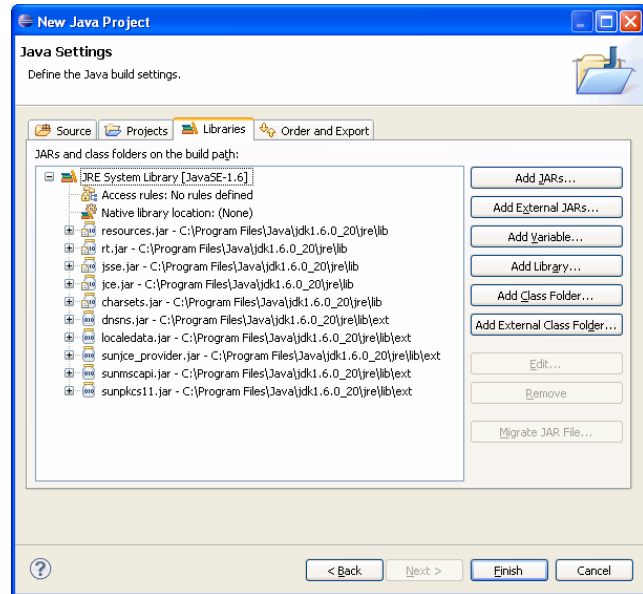
und Next >

*Project name* eingeben, z.B. *InstallTest*.



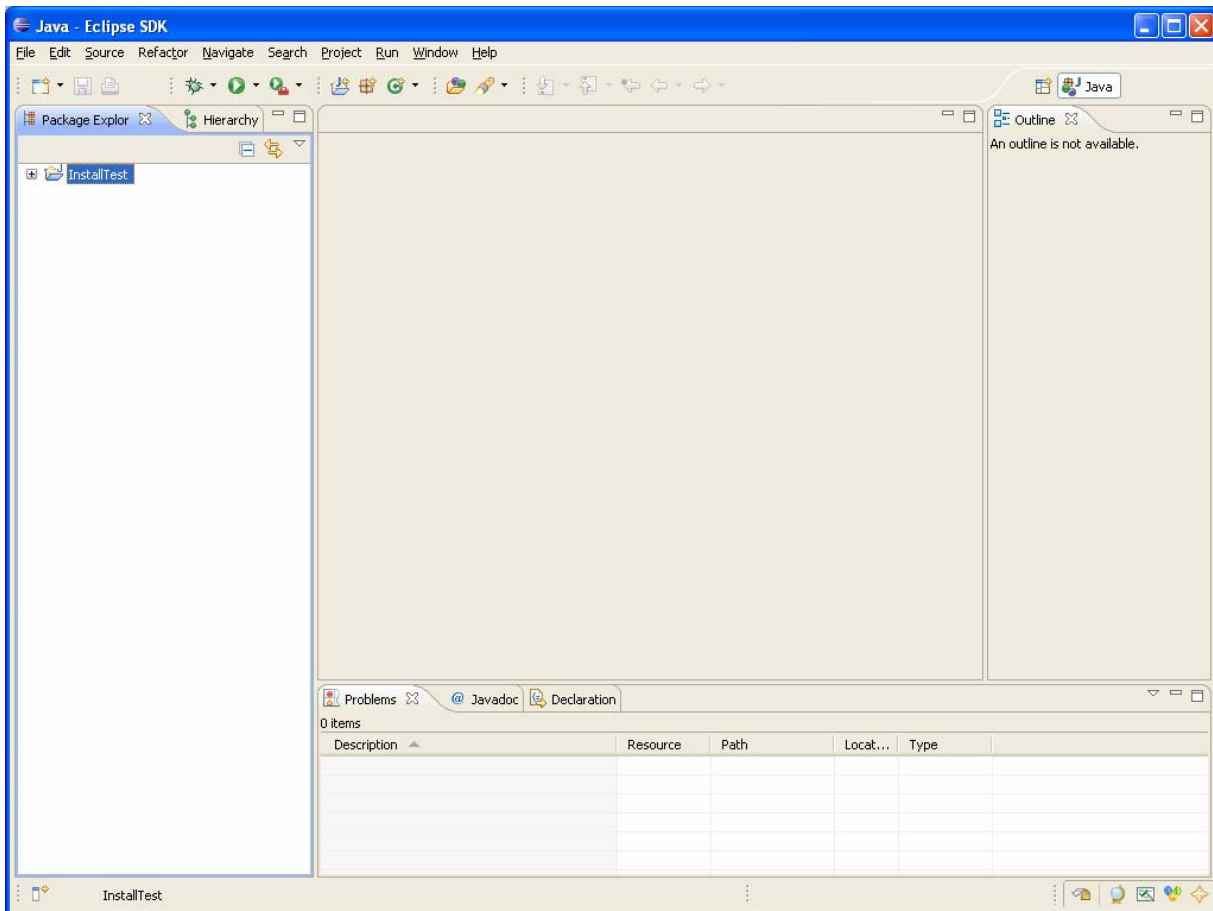
Next >

Im Tab *Libraries* die *Runtime-Library JRE* überprüfen:  
muss der installierten Java-Umgebung entsprechen.



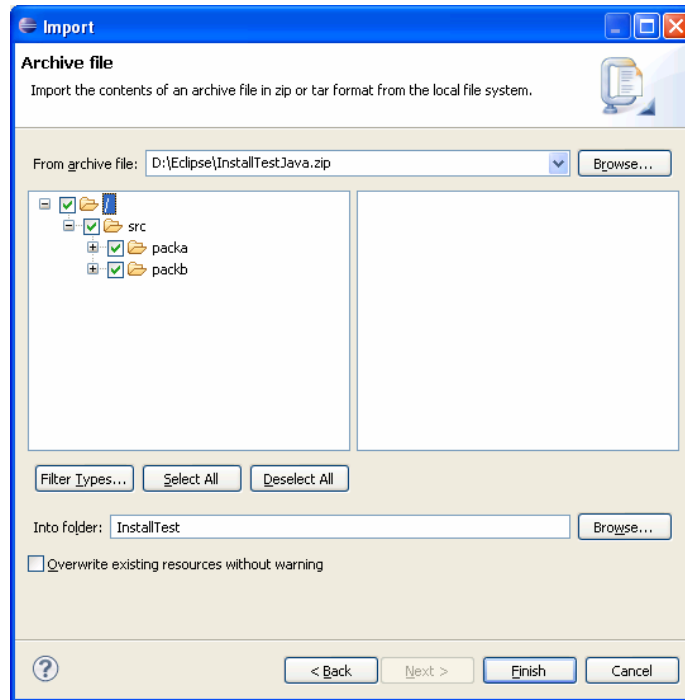
Finish

Im *Package Explorer* erscheint ein Ordner *InstallTest* :



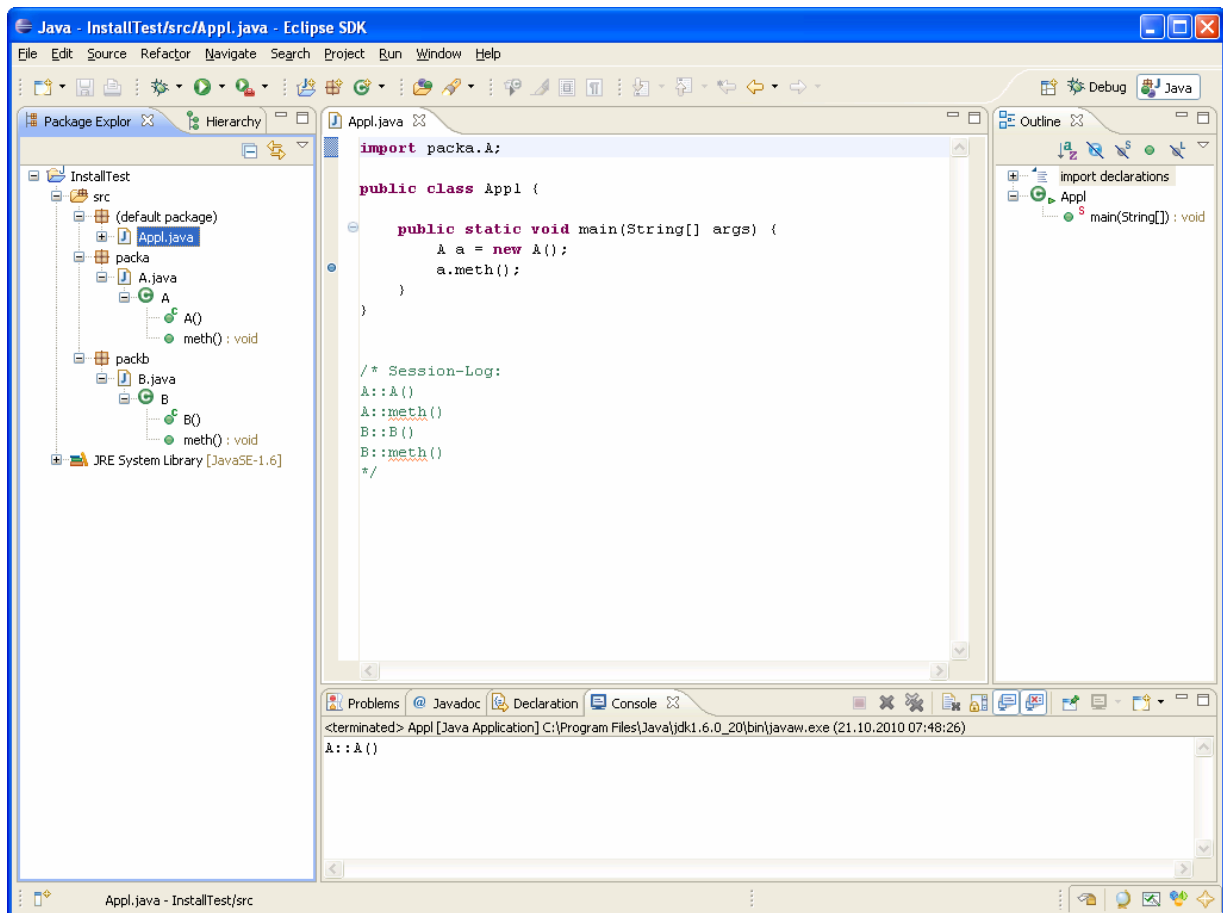
Die vorbereiteten Files aus *InstallTestJava.zip* in das neue Projekt importieren mit:

**Menü:File>Import... >  
General > Archive File :**



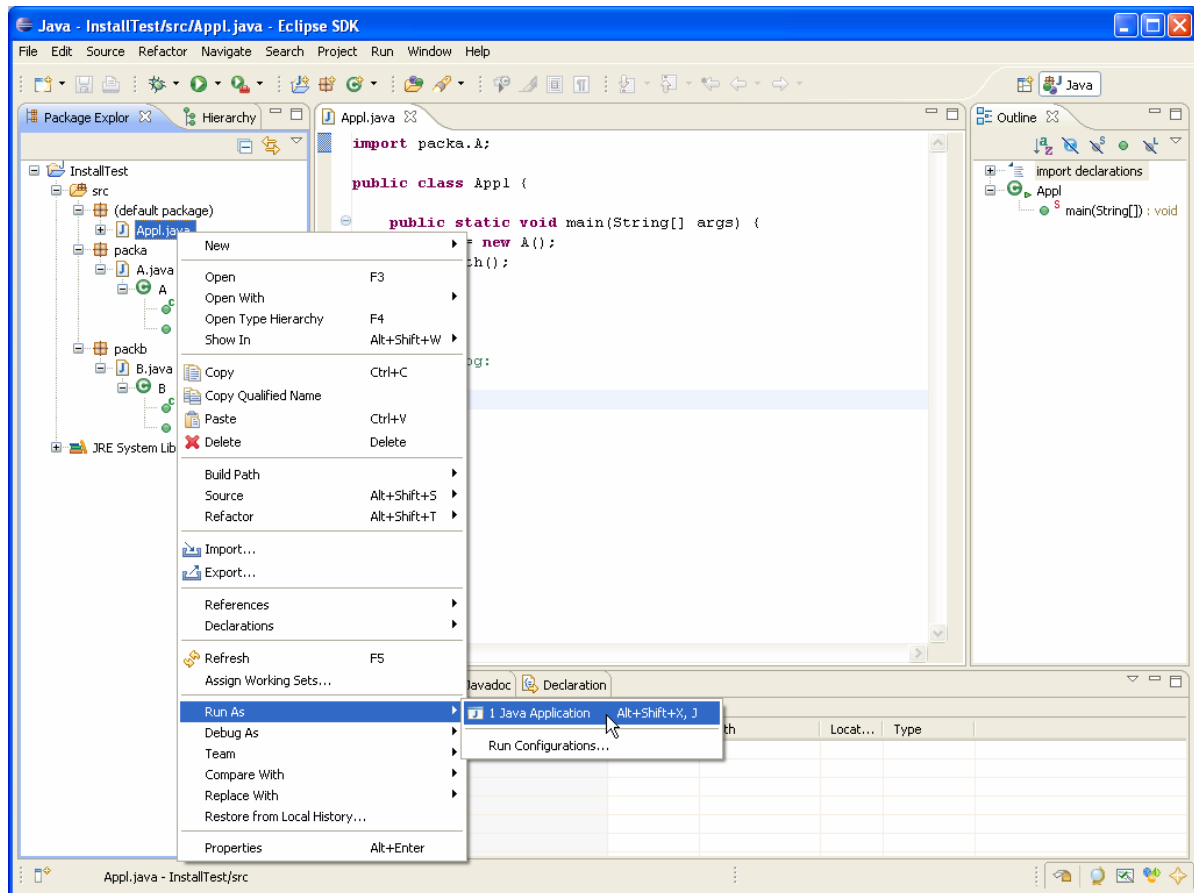
**Finish**

Zurück im Haupt-Window kann man im *Package Explorer* den Ordner *InstallTest* öffnen, dann *src* und darin wiederum z.B. die *Packages (default package, packa, packb)*, etc.. Durch *Doppel-Click* auf *Appl.java* wird das File geöffnet (mittlerer Bereich) und in der *Outline-View* (ganz rechts) wird die Struktur innerhalb des Files dargestellt.

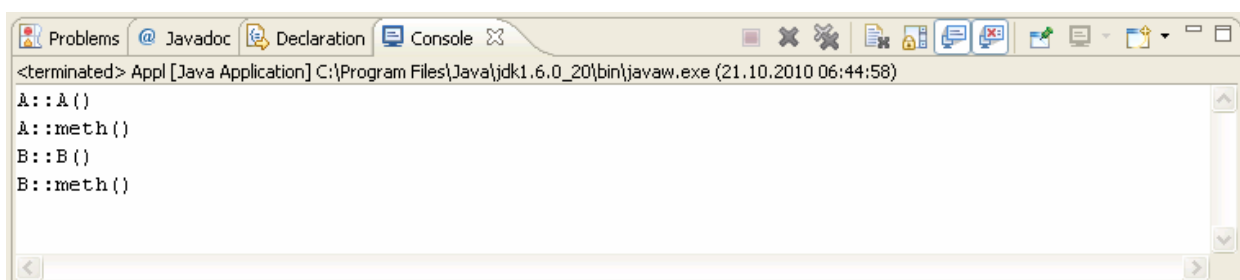


Die Applikation starten:

1. im *Package Explorer* das File *Appl.java* selektieren
2. Context-Menü (rechte Maustaste): *Run As > Java Application*



Rechts unten erscheint das neue Window *Console* (neben *Declaration*).  
Darin erscheint die *Standard-* und *Error-Ausgabe* des Programmes:



Von nun an kann die Applikation direkt mittels des Button  in der Toolbar gestartet werden.

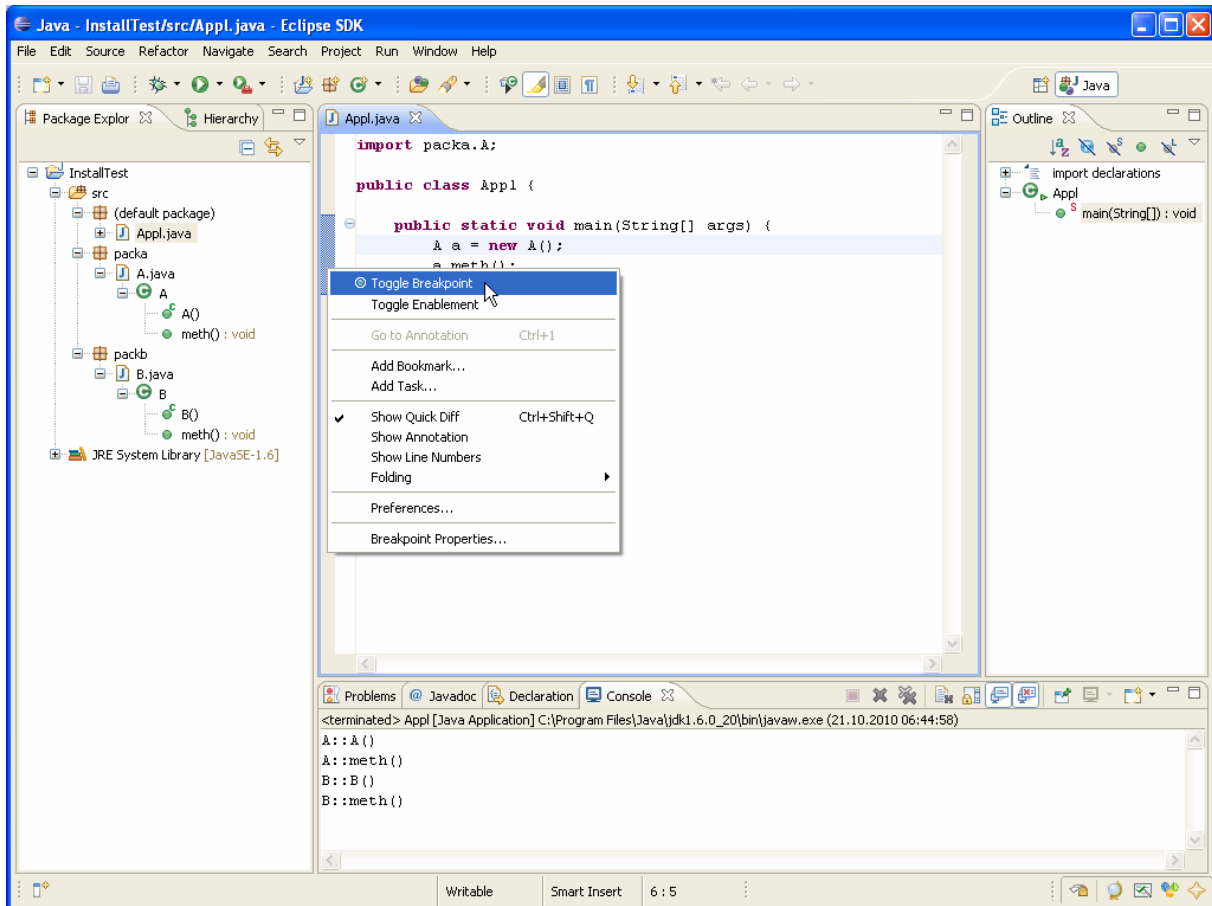
(Hinweis: das bedeutet also, dass *Run>Run As>Java Application* nur *genau einmal* durchgeführt werden muss!)



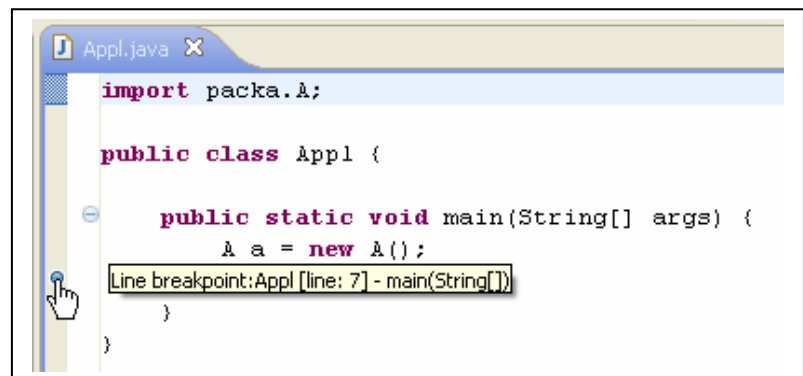
## 4 Debugging


Hinzufügen eines **Breakpoints** (*Haltepunkt*), z.B. auf der Zeile `a.meth()`:

Rechte Maustaste im linken Balken des Text-Editors des Files `Appl.java`: **Toggle Breakpoint**



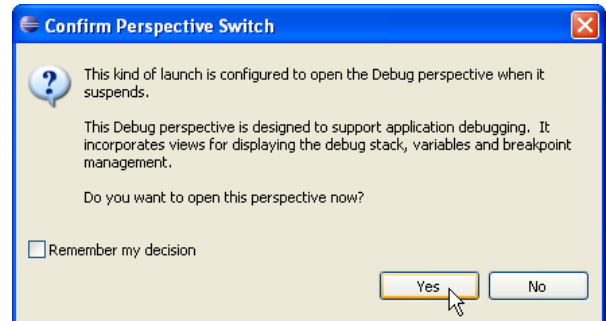
Danach ist der Breakpoint als *blauer Punkt* im Balken sichtbar:



Starten des Debuggers mittels des *Wanzen*(*Bug* ;-)-Button  in der Toolbar.

Ein Dialog erscheint mit der Frage, ob eine neue Perspektive *Debug* geöffnet werden soll.

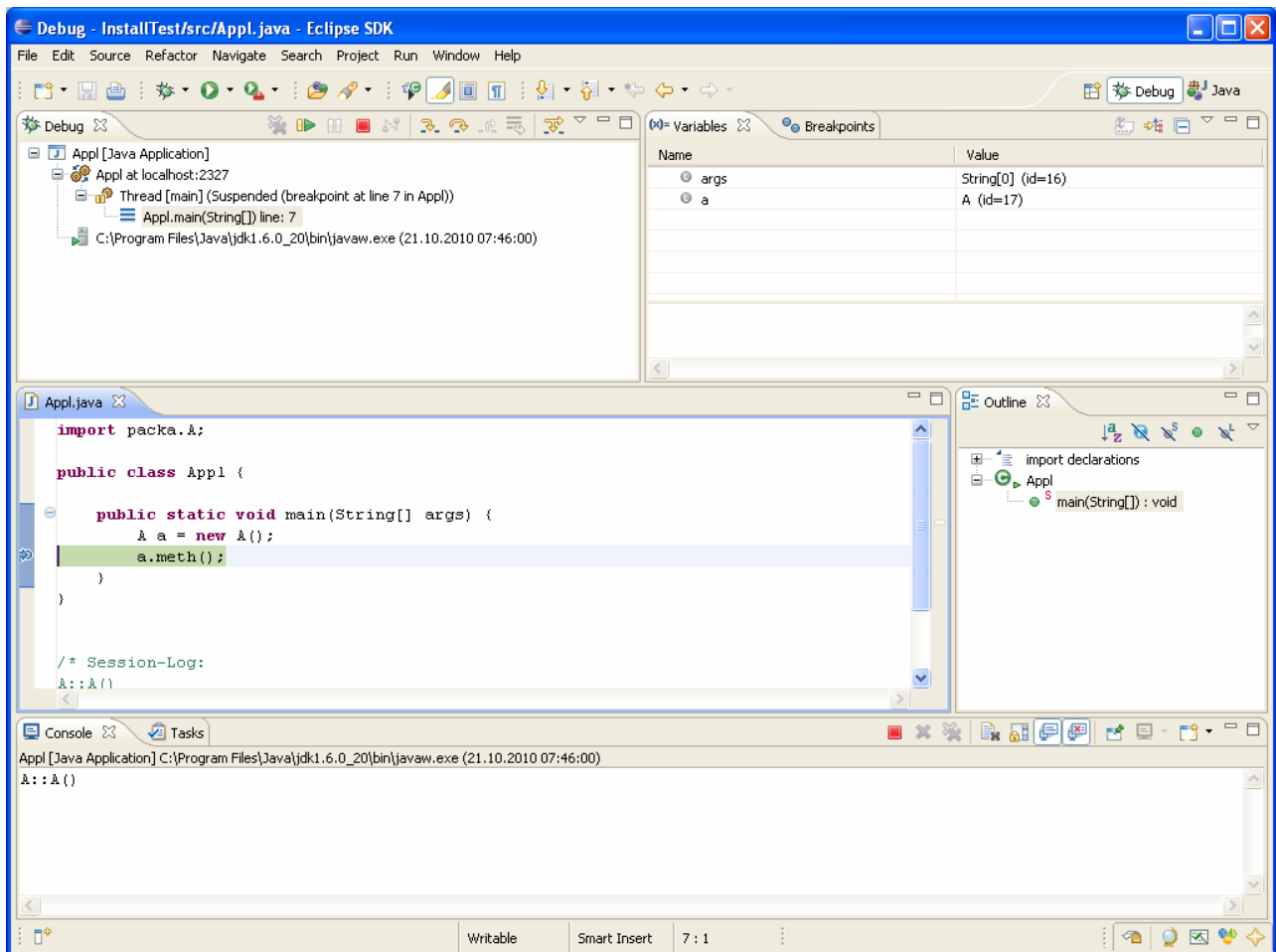
*Yes*



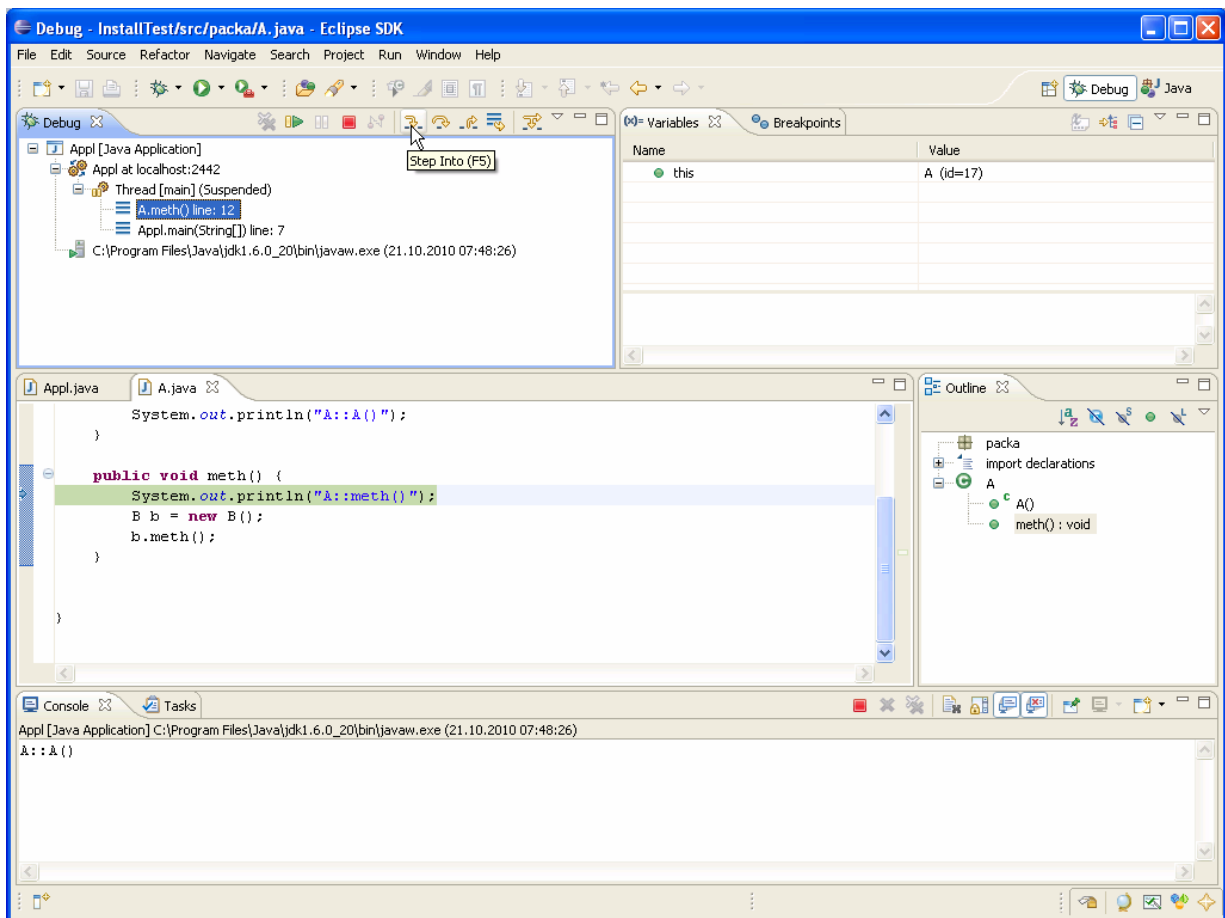
Eventuell muss *Eclipse* im *Firewall* freigeschaltet werden

Die neue Perspektive *Debug* wird geöffnet.

Das Programm läuft bis zum *Breakpoint* und bleibt dann stehen:

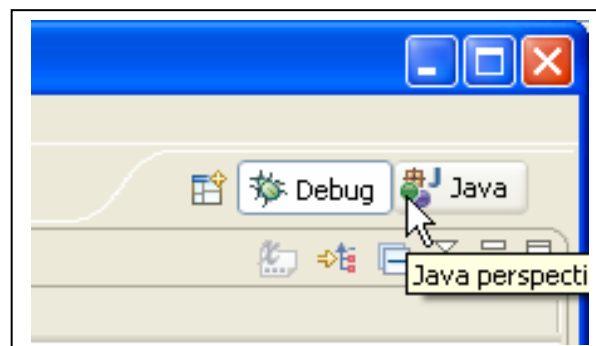


Mit **Run** > **Step Into** oder **Funktionstaste F5** oder das entsprechende Icon in der Toolbar kann das Programm schrittweise abgearbeitet werden:



Der *Debugger* kann mit **Menü: Run** > **Terminate** abgebrochen werden.

Mit Maus-Click auf den Button *Java* kommt man wieder zurück zur normalen Java-Perspektive:



*Congratulation! Everything o.k. - Good Bye ;-)*