

# Tutorial 14

## Eclipse

### 1. Einführung

#### 1.1 Übersicht

Es ist ausgesprochen unüblich, die Entwicklung neuer Anwendungen auf dem gleichen Server zu betreiben, auf dem die Anwendungen später einmal laufen sollen. In einem Unternehmen unterscheiden wir in der Regel zwischen dem/den Rechner(n) einer Entwicklungsumgebung, auf denen neue Anwendungen entstehen, und dem/den Rechner(n) einer Produktionsumgebung, auf denen die Anwendung später einmal laufen sollen.

Entwicklungen erfolgen meistens im Team. Die Entwickler haben jeder einen eigenen leistungsfähigen Arbeitsplatzrechner, auf dem editiert, kompiliert und getestet wird. Alle Arbeitsplatzrechner sind mit einem gemeinsamen Entwicklungsserver, dem Repository, verbunden, auf dem Quellcode und Maschinencode in unterschiedlichen Versionen zentral gespeichert wird. Auf dem Repository-Rechner laufen Funktionen wie die zentrale Steuerung des Entwicklungsprojektes, Versionsverwaltung, Backup-Verwaltung, und der Integrations- und Systemtest der von den einzelnen Entwicklern erstellten Komponenten.

Nach Abschluss einer Entwicklung wird der so entstandene Code in die Produktionsumgebung exportiert.

(Wir vernachlässigen an dieser Stelle die Tatsache, dass in der Regel nicht in die Produktionsumgebung, sondern in eine Testumgebung exportiert wird, die in Bezug auf Hard- und Software der Produktionsumgebung möglichst ähnlich ist).

Die Software der Entwicklungsumgebung sieht ganz anders aus als die Software der Produktionsumgebung. Als Entwicklungsrechner werden Windows 2000-Rechner oder Linux-Rechner eingesetzt. Das Repository befindet sich in der Regel auf einem getrennten Unix- oder Windows 2000-Rechner.

Bei der Produktionsumgebung unterscheiden wir zwischen einer 2-Tier- und einer 3-Tier-Konfiguration. Bei der 2-Tier-Konfiguration laufen alle Anwendungen auf dem zentralen OS/390-Rechner. Bei der 3-Tier-Konfiguration läuft ein Teil der Anwendungen auf einem getrennten Unix- oder Windows 2000 Server; auf dem OS/390-Rechner befindet sich die zentrale Datenhaltung (z.B. DB2) und evtl. ein Teil der Anwendungen (z.B. CICS-Programme). Die Arbeitsplatzrechner der Benutzer der Produktionsumgebung haben in der Regel nur eine geringe Funktionalität. In vielen Fällen ist dies lediglich ein Browser.

Bild 1 zeigt diese Zusammenhänge.

Wir wollen diesen Entwicklungsprozess nachvollziehen. Es sollen zwei Java-Programme erstellt werden, welche entweder als normale .class Datei oder als Servlet auf eine OS/390 DB2-Datenbank zugreifen. Diese Java-Programme werden in einer Windows 2000-Entwicklungsumgebung erstellt und sollen entweder unter OS/390 oder auf einem Middle Tier-Rechner ausgeführt werden. Im letzteren Fall greifen sie über das Netz auf die OS/390 DB2-Datenbank zu.

Für den Datenbank-Zugriff existieren vorgefertigte Java-Klassen, die als „Connectoren“ bezeichnet werden. Wir verwenden zwei unterschiedliche Connectoren: JDBC und DB2Connect. JDBC ist ein universeller Connector für den Zugriff auf beliebige relationale Datenbanken. Die Universalität hat ihren Preis; das Leistungsverhalten kann unzureichend sein. DB2Connect ist für die DB2-Datenbank optimiert und in Produktionsumgebungen häufig die bessere Alternative.

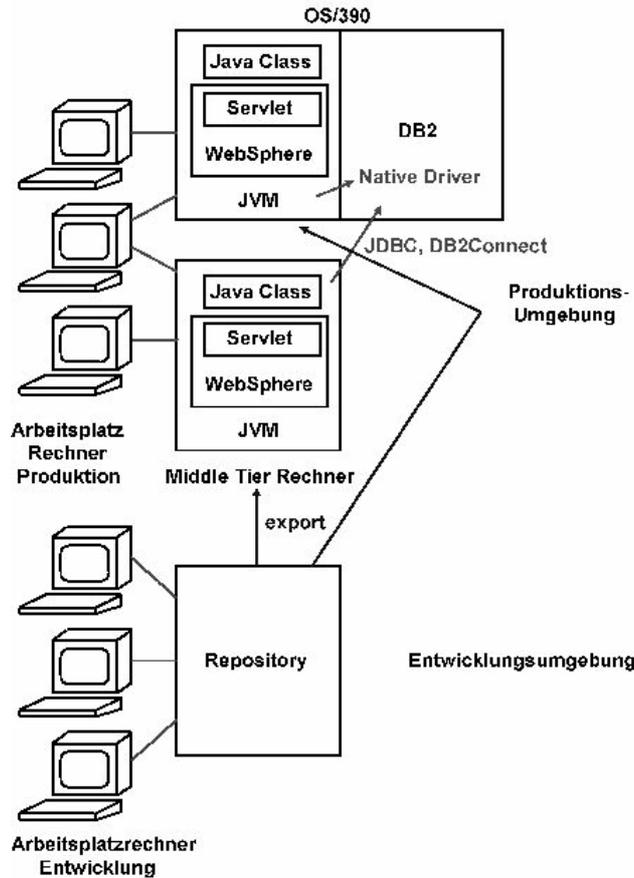


Bild 1 : Entwicklungs- und Produktionsumgebung

Als Entwicklungsumgebung (auch als Workbench oder als IDE – Integrated Development Environment - bezeichnet) verwenden wir ein Open Source-Produkt **Eclipse**. Vergleichbare Entwicklungsumgebungen existieren von vielen anderen Herstellern; Eclipse hat den Vorteil, dass es besonders einfach ist, neue Anwendungen mit OS/390 zu integrieren. Ein weiterer Vorteil ist, dass es von vielen Herstellern unterstützt wird.

Unsere .class Datei läuft in einer normalen Java Virtuellen Maschine (JVM). Unser Servlet benötigt eine Servlet-Laufzeitumgebung (Servlet Engine). Wir verwenden hierfür den public Domain **Tomcat** Web Application Server.

Das vorliegende Tutorial behandelt einen JDBC-Zugriff auf die DB2-Datenbank des OS/390-Rechners jedi.informatik.uni-leipzig.de mit Hilfe eines JDBC Connectors. Es wird durch ein weiteres Tutorial ergänzt, welches den Zugriff über DB2Connect durchführt. Die in den beiden Tutorials verwendete Konfiguration besteht aus einem Windows 2000-Rechner, der über das Internet auf unseren OS/390-Rechner Jedi.informatik.uni-leipzig.de zugreift. Der Windows 2000-Rechner dient sowohl als Entwicklungsumgebung als auch als Middle Tier in einer 3-Tier-Konfiguration. Auf ihm ist Eclipse installiert. Für den Middle Tier-Einsatz verfügt Eclipse über eine Tomcat Test-Umgebung; eine getrennte Servlet Engine ist deshalb nicht erforderlich.

***Aufgabe:** Arbeiten Sie das nachfolgende Tutorial durch. Installieren Sie - wenn notwendig -, wie in diesem Tutorial beschrieben, alle notwendige Software. Diese befindet sich auf zwei CD und auf einem Server zum Download.*

## 1.2 Aufgabenstellung

Die beiden vorliegenden Tutorials (Tutorials 14 und 15) ermöglichen es, die Infrastruktur für eine e-Business Anwendungsumgebung einzurichten. Die Präsentationslogik und Businesslogik ist in einem Servlet enthalten, welches auf einem Windows 2000-Rechner läuft und Daten in einem Browserfenster wiedergibt. Das Servlet greift über Konnektoren und das Internet auf eine DB2-Datenbank zu, die auf dem OS/390-Rechner jedi.informatik.uni-leipzig.de vorhanden ist.

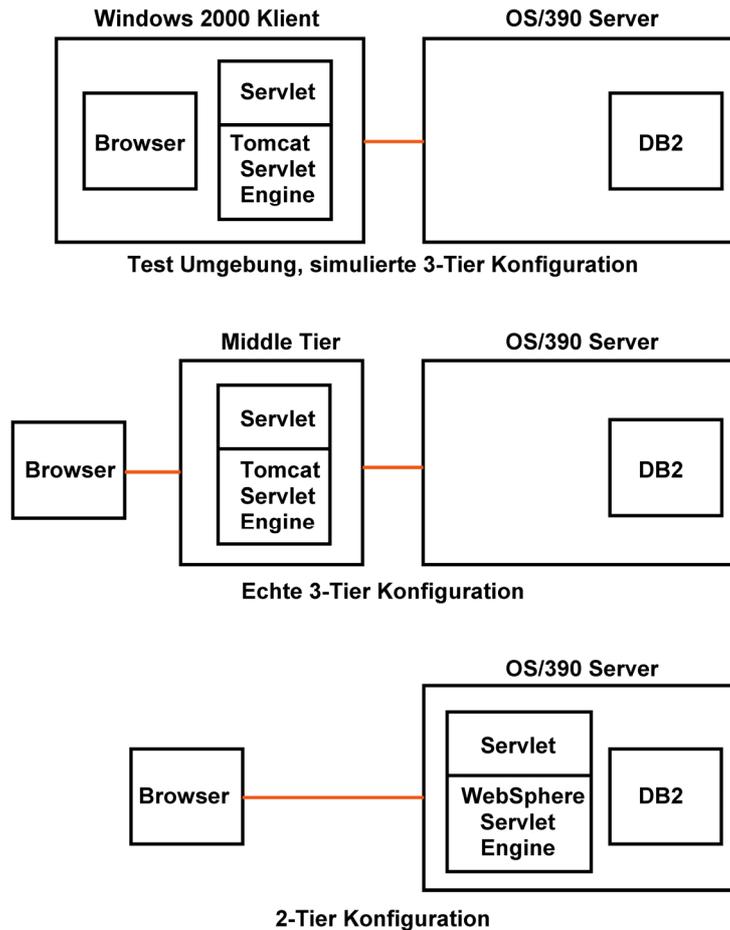


Bild 2

Die oberste der drei Abbildungen in Bild 2 zeigt die Testumgebung, die in diesem und dem nächsten Tutorial entwickelt wird. Hierbei greift das Servlet über einen JDBC-Konnektor (Hit Driver) auf die DB2-Datenbank zu. Diese Konfiguration ist leicht zu einer echten 3-Tier-Konfiguration erweiterbar, bei der unser Windows 2000-Entwicklungsrechner die Aufgabe des Middle-Tiers übernimmt (mittlere der drei Abbildungen).

In der untersten der drei Abbildungen ist eine 2-Tier-Alternative dargestellt. Hierbei dient der Windows 2000-Rechner lediglich als Entwicklungssystem. Nach Fertigstellung und abgeschlossenem Austesten wird das Servlet auf den OS/390-Server portiert, wo es als .class File unmittelbar ausführbar ist. Dieser Prozess wird in Tutorial 14 beschrieben.

Bild 3 zeigt diese Zusammenhänge in mehr Details.

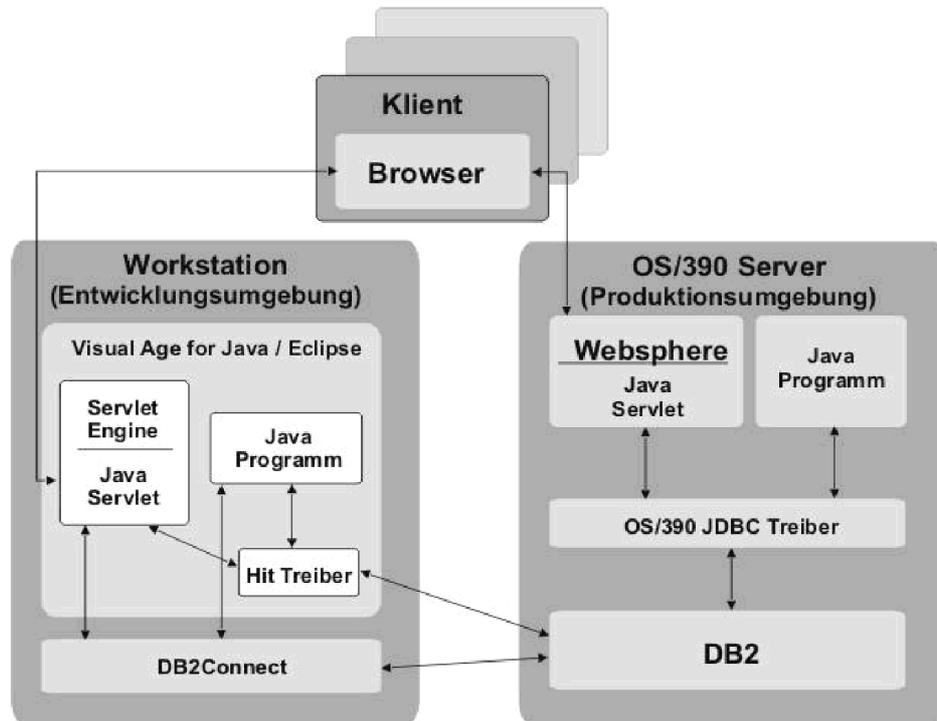


Bild 3

### 1.3 Installationsübersicht

Für die Installation sind die folgenden Schritte erforderlich:

1. Einführung (dieser Abschnitt)
2. JDK Installation
3. Eclipse Installation
4. Eclipse konfigurieren
5. **Eclipse** in Betrieb nehmen
6. **Eclipse** herunterfahren
7. Erneute Inbetriebnahmen nach abgeschlossener Installation
8. Installation von DB2 UDB 6.1
9. Installation von DB2Connect
10. Konfigurieren von DB2Connect
11. DB2Connect in Eclipse Integrieren
12. DB2Connect Verbindung in Betrieb nehmen

Das vorliegende Tutorial 14 beschreibt Schritte 1 – 7. Schritte 8 – 12 sind in einem weiteren Tutorial 15 enthalten.

### 1.4 Installations-Voraussetzungen und Vorbereitung

Diese Installation nimmt an, dass sich auf einem (möglicherweise Multiboot-) Rechner eine Partition H: befindet, auf der eine frische Kopie von Windows 2000 Professional Edition installiert ist. Wird eine andere Partition benutzt, muss der entsprechende Buchstabe anstelle von H: benutzt werden. Dieser Rechner soll als Klient für den OS/390-Server [jedi.informatik.uni-leipzig.de](http://jedi.informatik.uni-leipzig.de) dienen.

Einloggen erfolgt für den Benutzer *administrator* mit Administrator Rechten (wichtig !)

Ausgangsbasis sind zwei CDs. In der hier beschriebenen Beispielsinstallation wurden beide CDs auf der Festplatte in der Partition F: zwischengespeichert. Es ist aber auch möglich, die Installation direkt von den CDs durchzuführen. Die erste der beiden CDs enthält die Verzeichnisse Eclipse und

UDB\_6.1 ; die zweite der beiden CDs enthält die Verzeichnisse DB2Connect und das (hier nicht benötigte) Verzeichnis db2fix.

Auf der CD # 1 befinden sich im Verzeichnis Eclipse die folgenden Dateien (insgesamt circa 100 Mbyte):

```
<DIR>          diplom
 58.475.129 eclipse-SDK-I20021018-win32.zip
          117 hallo.java
          245.963 hit.jar
37.114.037 j2sdk-1_4_0_02-windows-i586.exe
 4.622.570 jakarta-tomcat-4.0.4.exe
          168.316 tomcatPluginV201.zip
```

Wir wechseln in das Verzeichnis H:\Programme und erstellen dort ein neues Unterverzeichnis H:\Programme\jdk. Wir wechseln in dieses Unterverzeichnis und kopieren von der CD # 1, Verzeichnis eclipse, das Programm

j2sdk-1\_4\_0\_02-windows-i586.exe

dort hinein. Wir wechseln in das Root-Verzeichnis H:\ und erstellen dort ein neues Verzeichnis H:\Eclipse. Wir wechseln in dieses Unterverzeichnis und kopieren von der CD alle Dateien sowie den Unterordner des Ordners eclipse in dieses Unterverzeichnis hinein. Da wir j2sdk-1\_4\_0\_02-windows-i586.exe bereits in ein anderes Verzeichnis kopiert haben, muß diese Datei nicht kopiert werden. Wir können sie auch mitkopieren und anschließend wieder löschen.

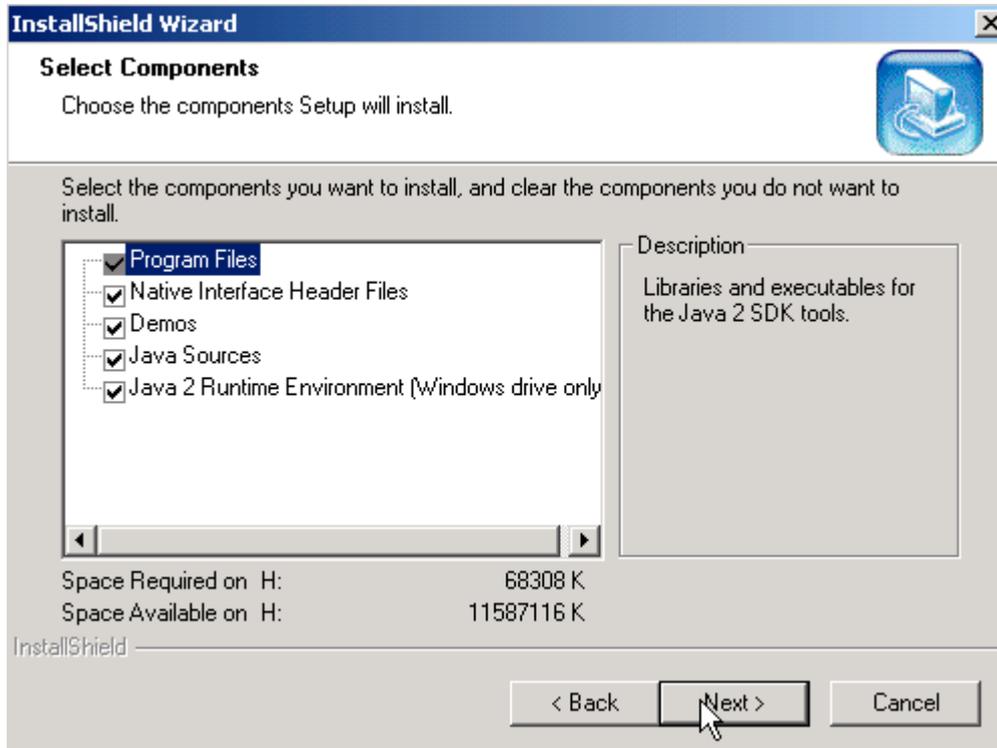
Es werden die folgenden Abkürzungen verwendet:

- 1k bedeutet einmal klicken mit der linken Maustaste
- 2k bedeutet zweimal klicken mit der linken Maustaste (Doppelklick)
- 1kr bedeutet einmal klicken mit der rechten Maustaste
- 2kr bedeutet zweimal klicken mit der rechten Maustaste (Doppelklick)

## 2. JDK-Installation

Windows Explorer aufrufen. In das Verzeichnis H:\programme\jdk wechseln.

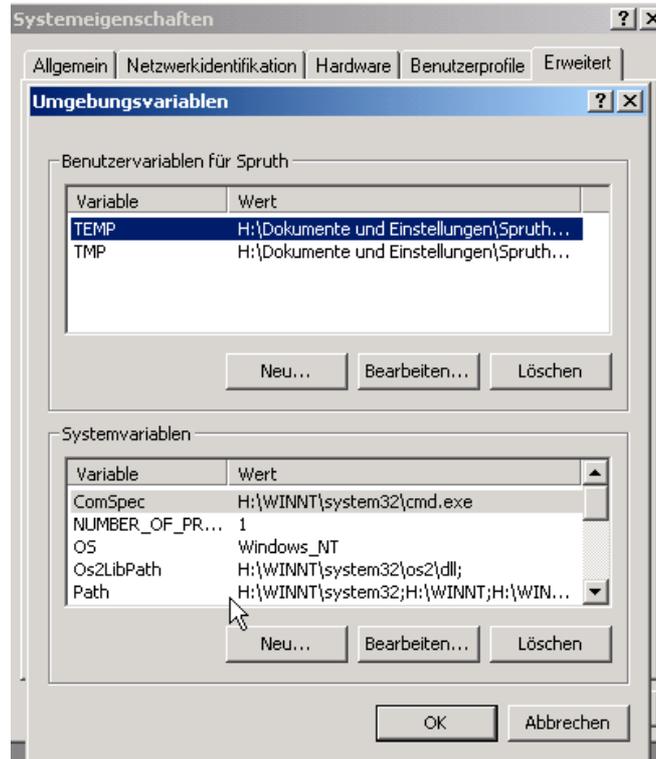
Mit der linken Maustaste Doppelklick auf j2sdk-1\_4\_0\_02-windows-i586.exe  
→ Next → yes → destination Folder "H:\j2sdk1.4.0\_02 → Next



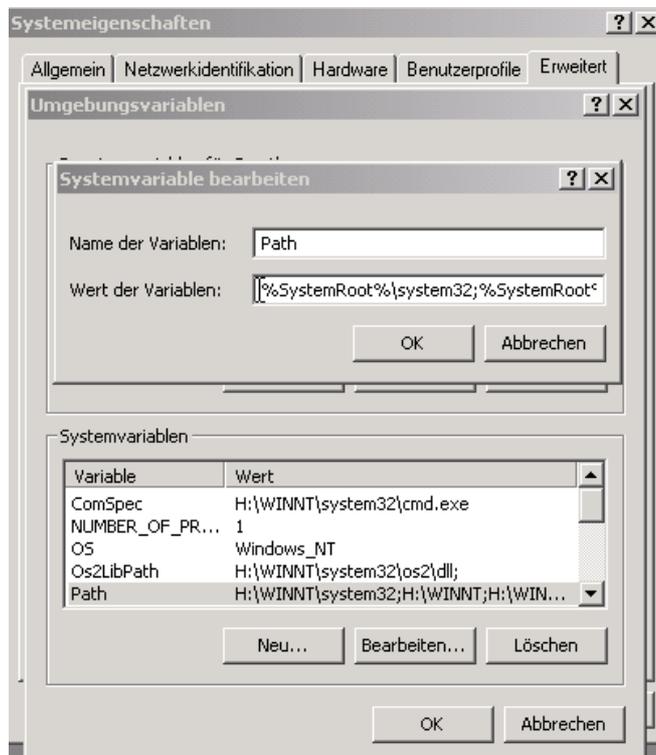
Next → select Microsoft Internet Explorer → Next → warten, setting up Java Runtime Environment, warten → Finish.

System neu starten.

Start → Einstellungen → Systemsteuerung; System 2k, Erweitert 1k, Umgebungsvariablen 1k

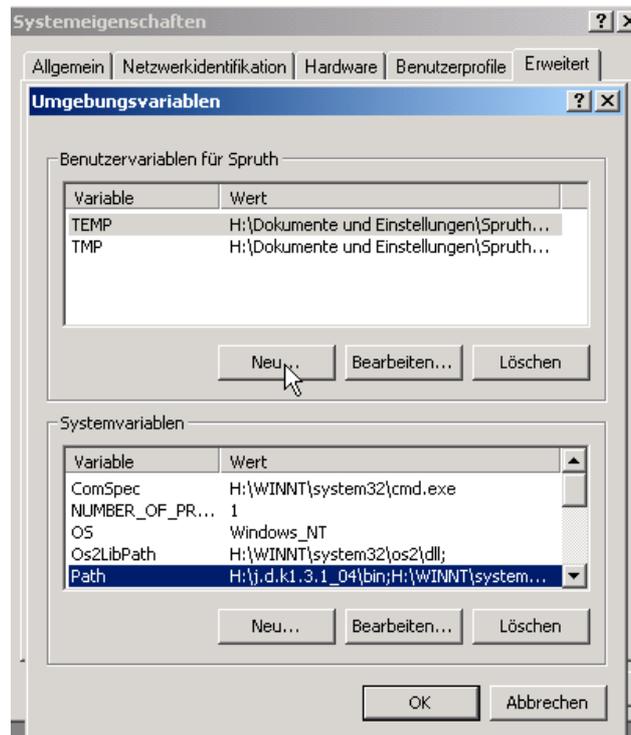


Im Fenster Systemvariablen mit dem Cursor auf die Zeile *Path* 2k (oder alternativ 1k auf die Zeile *Path* und 1k auf *Bearbeiten*)

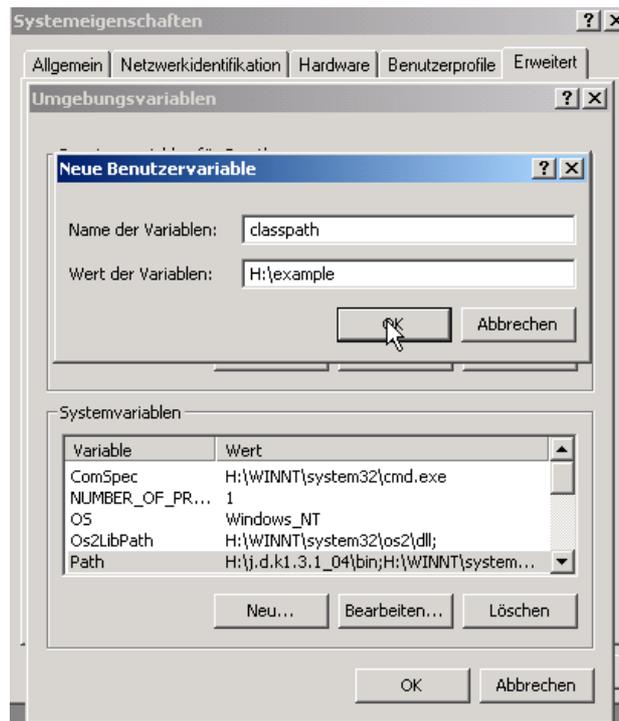


Das Fenster *Systemvariable bearbeiten* erscheint. Mit dem Cursor in das Feld Wert der Variablen gehen, 1k. Mit der Pfeiltaste "←" ganz nach links scrollen (oder alternativ die Taste Pos1 drücken).

Vor %SystemRoot den Wert H:\j2sdk1.4.0\_02\bin; eingeben;  
1k auf ok im Fenster *Systemvariable bearbeiten*.



Im Fenster *Benutzervariablen für Administrator 1k* auf Neu;



Jetzt folgendes eingeben:

Name der Variablen: classpath  
Wert der Variablen: H:\example

Klick auf OK schließt das Fenster *Neue Benutzervariable*.

Ein weiterer Klick auf den Button OK des Fensters *Umgebungsvariablen* schließt dieses Fenster.

Ein dritter Klick auf den Button OK des Fensters *Systemeigenschaften* schließt auch dieses Fenster.

Schließen Sie alle restlichen Fenster, die noch offen sind und starten Sie den Rechner neu. Hiermit werden die neuen Umgebungsvariablen wirksam.

Legen Sie jetzt ein neues Verzeichnis `H:\example` an.

Mit einem beliebigen Editor das folgende Programm erzeugen:

```
class hallo {  
    public static void main(String[] args) {  
        System.out.println("Hallo Welt really");  
    }  
}
```

Dieses Programm als Datei `hallo.java` im Verzeichnis `H:\example` speichern. Eine Kopie dieser Datei befindet sich bereits im Verzeichnis `H:\Eclipse`.

In einem DOS-Fenster in das Verzeichnis `h:\example` wechseln.

Wir übersetzen das Programm mit dem Befehl `javac hallo.java` Kurz warten. Wenn keine Rückmeldung erfolgt (mit Ausnahme des Promptes), war die Übersetzung erfolgreich. Im Verzeichnis `H:\example` steht jetzt die Datei `hallo.class` . Jetzt `java hallo` eingeben. Sie sollten die Meldung

**Hallo Welt really** sehen.

Damit wurde der JDK erfolgreich installiert.

### 3. Eclipse-Installation

Wenn noch nicht auf ihrem PC vorhanden, laden Sie Winzip von [www.winzip.com](http://www.winzip.com) herunter und installieren Sie es.

Windows Explorer aufrufen. In das Verzeichnis H:\Eclipse gehen (in das vorher die Dateien von der CD kopiert wurden). 2k auf eclipse-SDK-I20021018-win32.zip



eclipse-SDK-I20021018-win32.zip

In das Verzeichnis H:\programme entpacken. Hierbei wird automatisch ein neues Verzeichnis H:\programme\eclipse angelegt, in welches die Dateien geladen werden.

```
<DIR>          features
<DIR>          plugins
<DIR>          readme
                61 .eclipseproduct
            15.048 cpl-v10.html
            86.016 eclipse.exe
                633 install.ini
                4.743 notice.html
            16.496 startup.jar
```

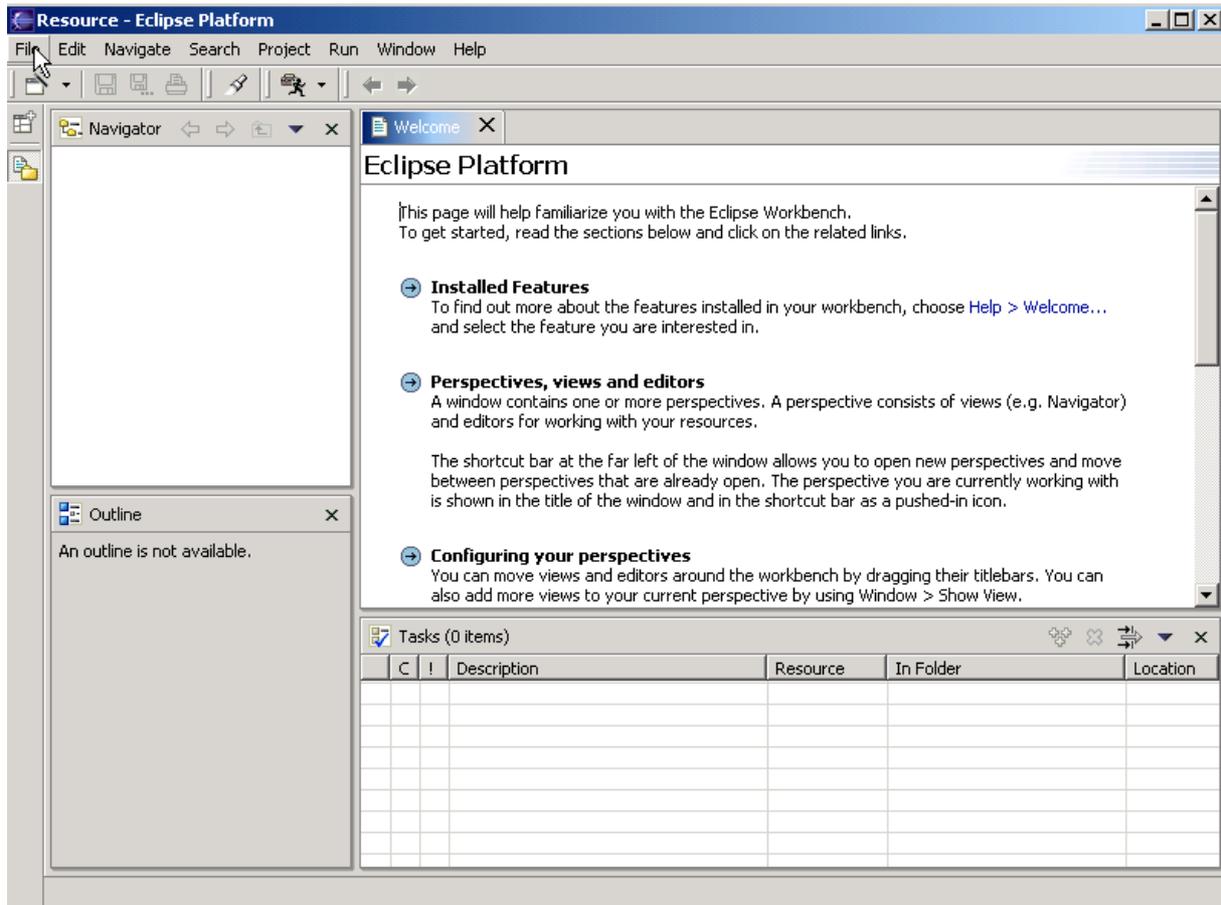
Inhalt von H:\programme\eclipse

**Schritt 1: Eclipse starten.** Mit dem Windows Explorer in das Verzeichnis H:\programme\eclipse gehen. 2k auf das folgende Symbol

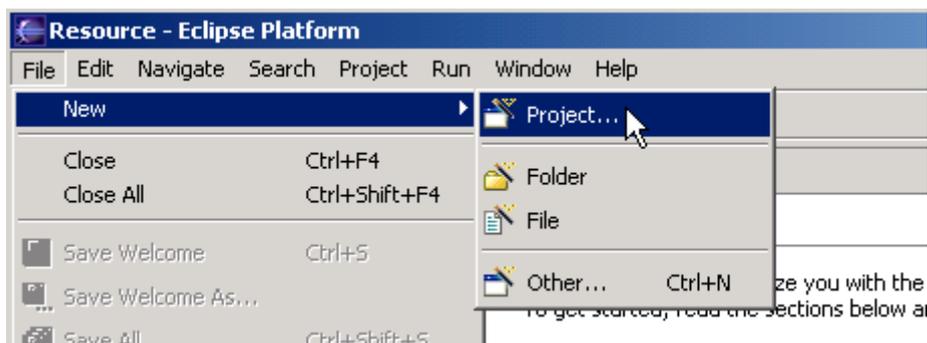


eclipse

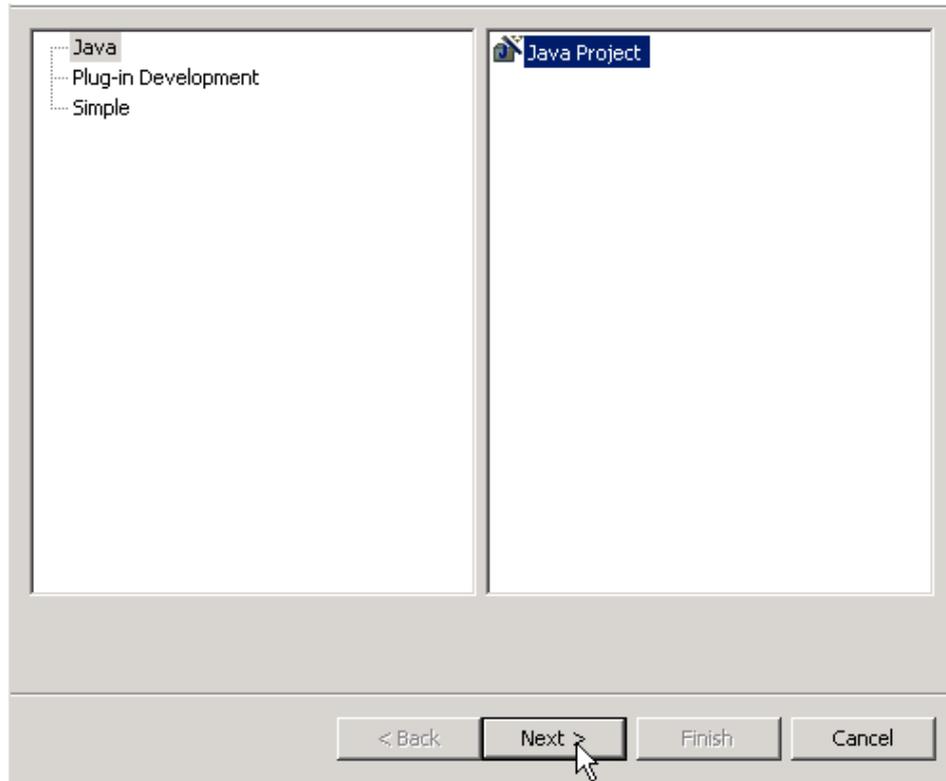
**Schritt 2: Projekt anlegen.** Nach einiger Zeit erscheint der Eclipse Platform Screen. Wir klicken links oben auf File, dann auf New und anschließend auf Project (siehe auch die nächsten zwei Abbildungen einschließlich deren Unterschriften).



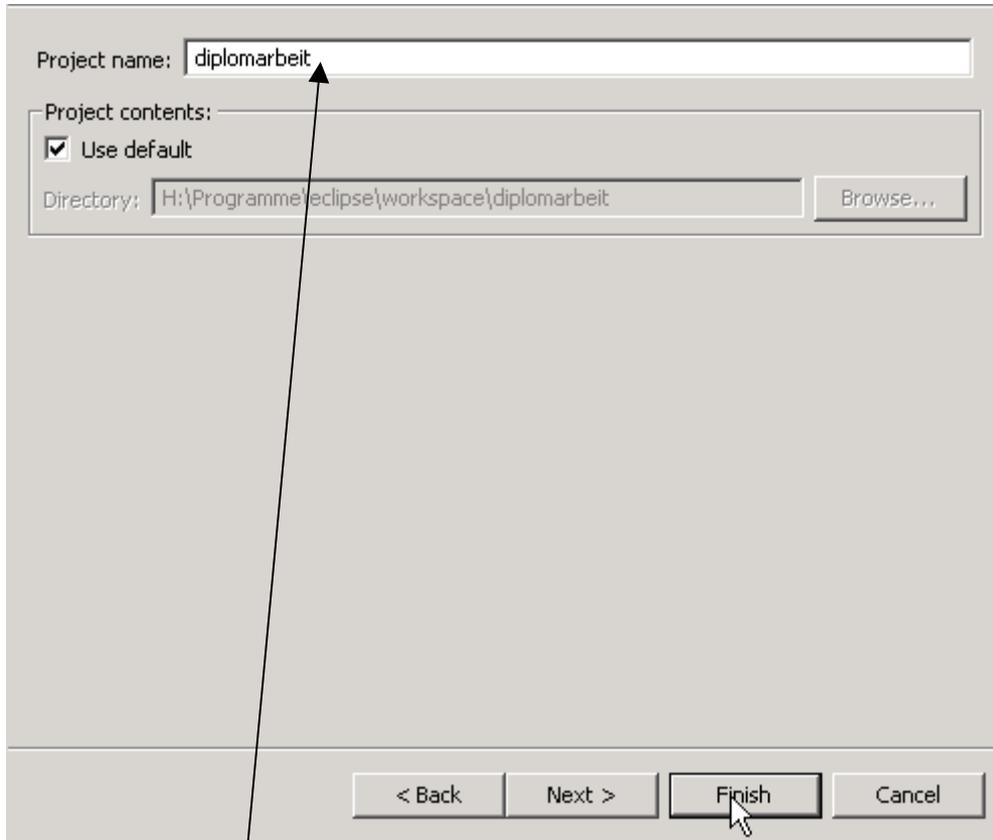
1k auf File,



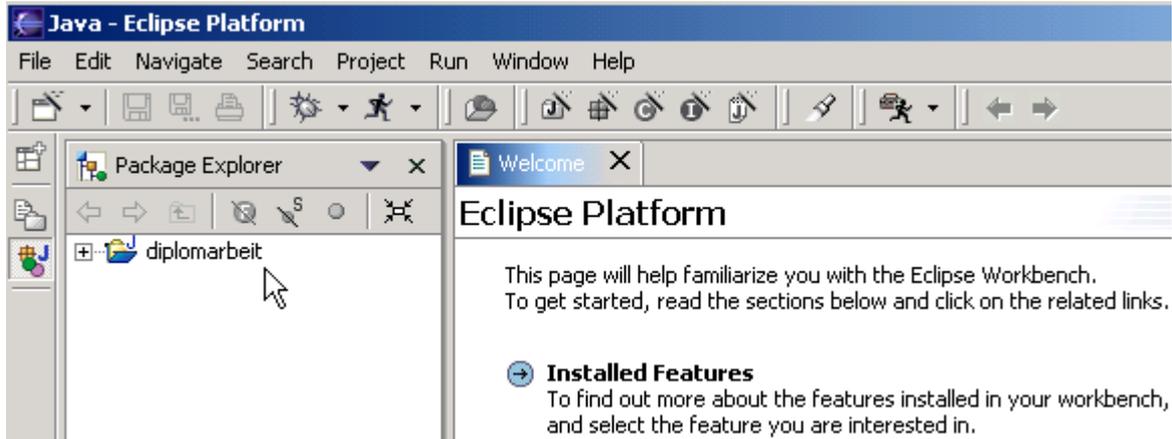
1k New, 1k auf Project,



1k auf *Java Project* (markieren), auf *Next* klicken.

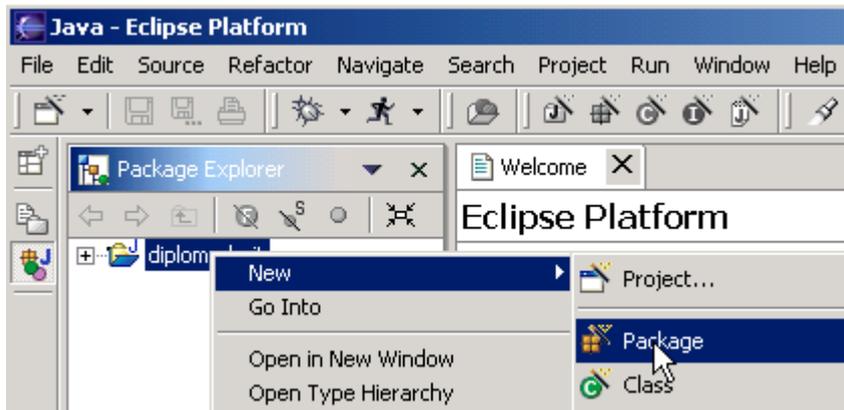


Project Name *diplomarbeit* eintippen (alles klein schreiben), Klick auf *Finish*.

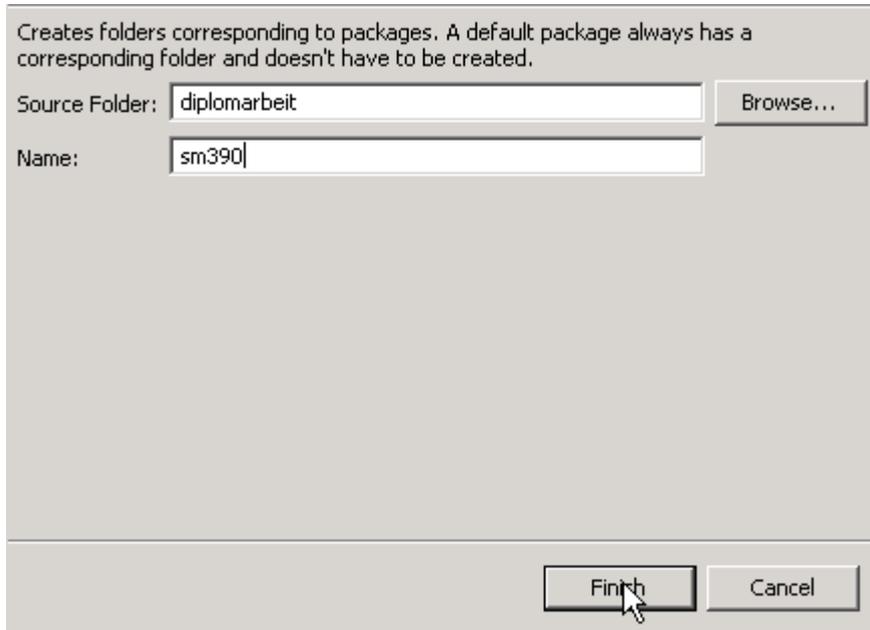


In dem linken Fenster ist jetzt der Eintrag *diplomarbeit* zu sehen.

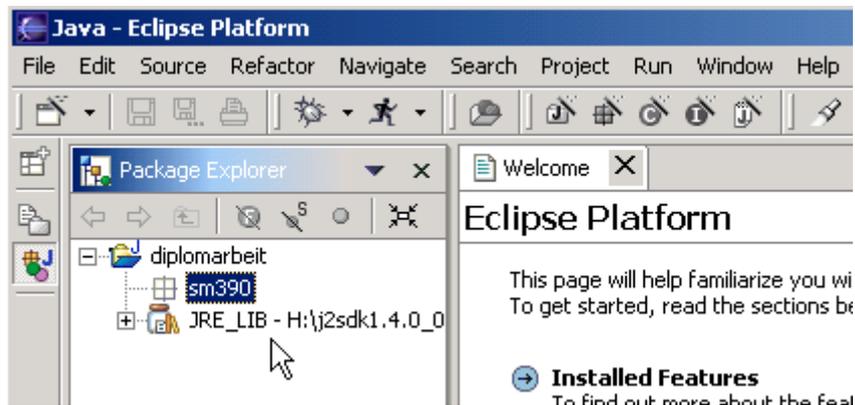
### Schritt 3: Sources importieren, Package anmelden



Auf "diplomarbeit" 1k, dann 1kr, 1k auf New, 1k auf Package.



Source Folder = *diplomarbeit* sowie Name = *sm390* eintippen, auf *Finish* klicken.



JRE wird von Eclipse alleine angelegt.

Nachfolgend der aktuelle Inhalt des Verzeichnisses *H:\Programme\eclipse*:

```

<DIR>      .
<DIR>      ..
<DIR>      features
<DIR>      plugins
<DIR>      readme
<DIR>      workspace
           61 .eclipseproduct
          15.048 cpl-v10.html
          86.016 eclipse.exe
           633 install.ini
           4.743 notice.html
          16.496 startup.jar
    
```

Ein neues Verzeichnis *H:\programme\eclipse\workspace* wurde angelegt. Dies ist der Inhalt:

```

H:\Programme\eclipse\workspace

<DIR>      .
<DIR>      ..
<DIR>      .metadata
<DIR>      diplomarbeit
    
```

```

H:\Programme\eclipse\workspace\.metadata

<DIR>      .
<DIR>      ..
<DIR>      .config
<DIR>      .plugins
           154 .cache.properties
           110 .keyring
            1 .lock
          344.897 .registry
    
```

```

H:\Programme\eclipse\workspace\diplomarbeit

<DIR>      .
<DIR>      ..
<DIR>      sm390
           247 .classpath
           388 .project
    
```

Das Verzeichnis *sm390* ist leer.

#### Schritt 4: Java-Quellcode importieren

Mit dem Windows Explorer in das Verzeichnis *h:\Eclipse\diplom* wechseln (dort stehen die von der CD kopierten Daten). Im Verzeichnis *Diplom* befindet sich der Java-Quellcode als .zip-Datei *source\_390.zip*. Diese Datei in das Verzeichnis *H:\programme\eclipse\workspace\diplomarbeit\sm390* entpacken. Im Verzeichnis *H:\programme\eclipse\workspace\diplomarbeit* finden wir die Dateien *.classpath* und *.project*. Im Verzeichnis *sm390* schließlich befinden sich die gerade entpackten Dateien *SMJDBCTestPrg.java* und *SMJDBCTestServlet.java* .

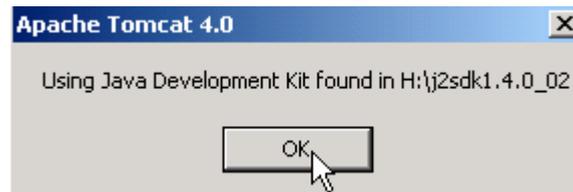
#### Schritt 5: Hit Driver installieren

Aus dem Verzeichnis *H:\Eclipse* die Datei *hit.jar* in das Verzeichnis *H:\programme\eclipse\workspace\diplomarbeit* verschieben oder kopieren.

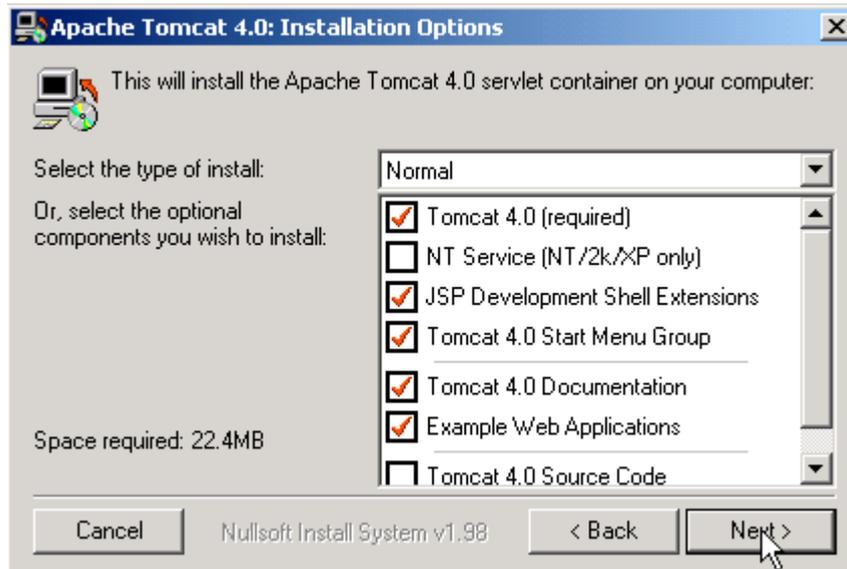
#### Schritt 6: Apache installieren

Mit dem Windows Explorer in das Verzeichnis *H:\Eclipse* wechseln (dort stehen die von der CD kopierten Daten).

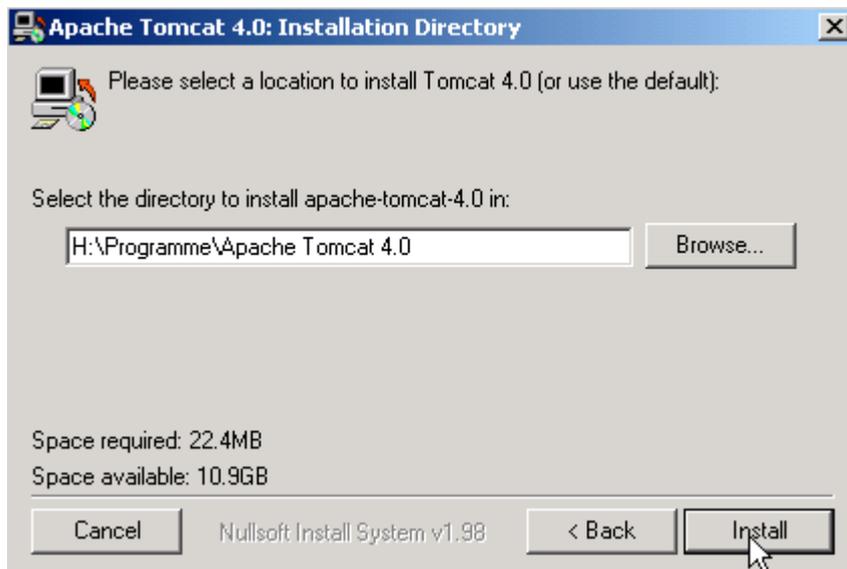
2k auf *jakarta-tomcat-4.0.4.exe*



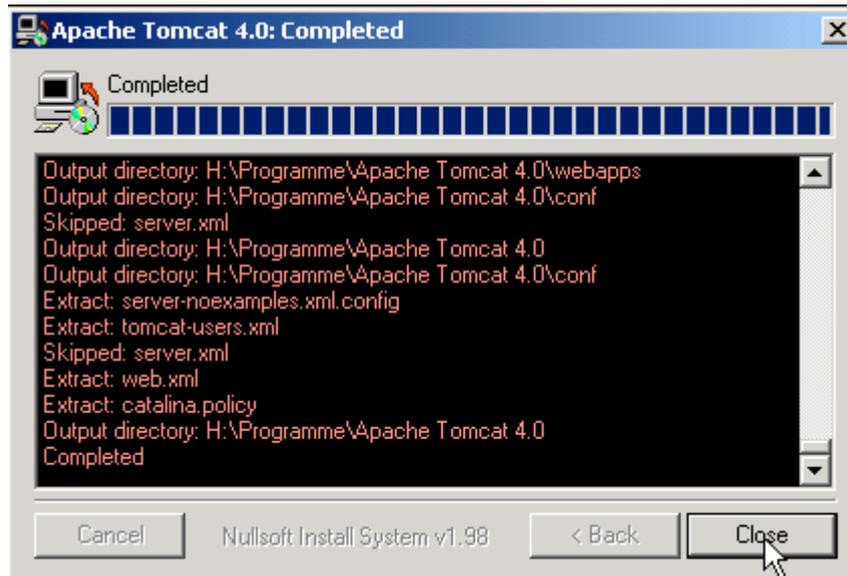
Erst auf *OK*, dann auf *I Agree* klicken.



Die vorgeschlagene Auswahl kann so akzeptiert werden. Keine Änderung. Auf *Next* klicken.



Das vorgeschlagene Verzeichnis *H:\programme\Apache Tomcat 4.0* akzeptieren. Keine Änderung. Auf den Button *Install* klicken – Daten werden übertragen.



Warten bis die Übertragung abgeschlossen ist und im Fenster die Meldung "Completed" erscheint. Auf den Button *Close* klicken.

### Schritt 7: Tomcat plugin in Eclipse installieren

Das Eclipse Platform-Fenster schließen.

Windows Explorer aufrufen. In das Verzeichnis *H:\Eclipse* gehen (in das vorher die Dateien von der CD kopiert wurden).

2k auf *tomcatPluginV201.zip*

Alle Files dieses Archives in das bereits existierende Verzeichnis *h:\programme\eclipse\plugins* extrahieren. Zu den bisher 64 Verzeichnissen wird ein 65. Verzeichnis *com.sysdeo.eclipse.tomcat* angelegt.

Winzip schließen.

### Schritt 8: Eclipse starten

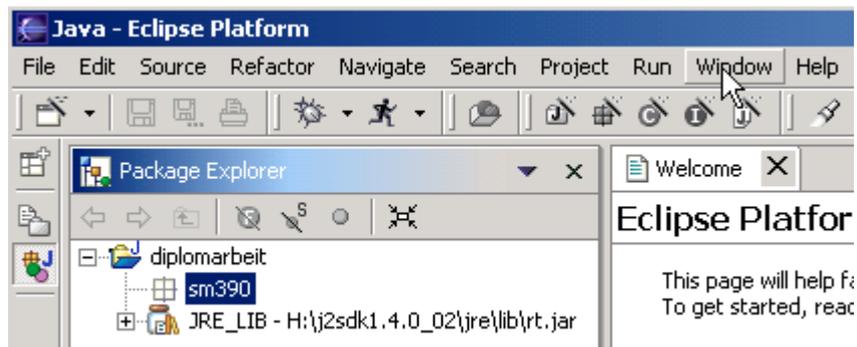
Windows Explorer aufrufen. In das Verzeichnis *H:\programme\eclipse* gehen.

2k auf das folgende Symbol:



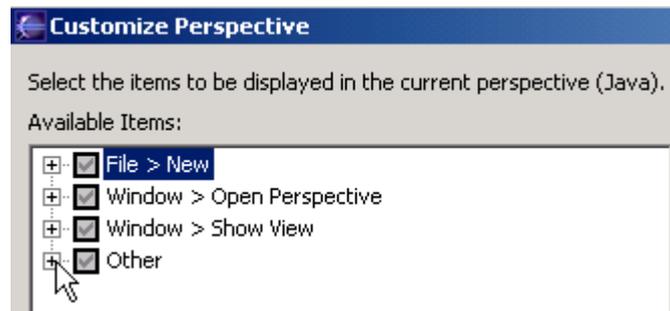
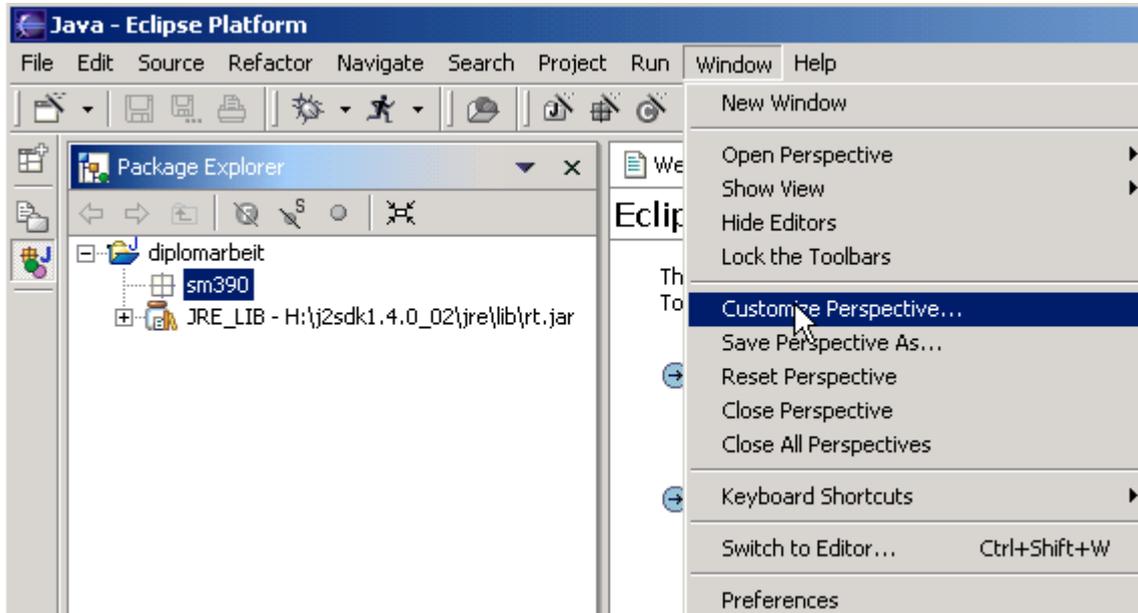
eclipse

Es dauert etwas, bis das folgende Fenster erscheint:

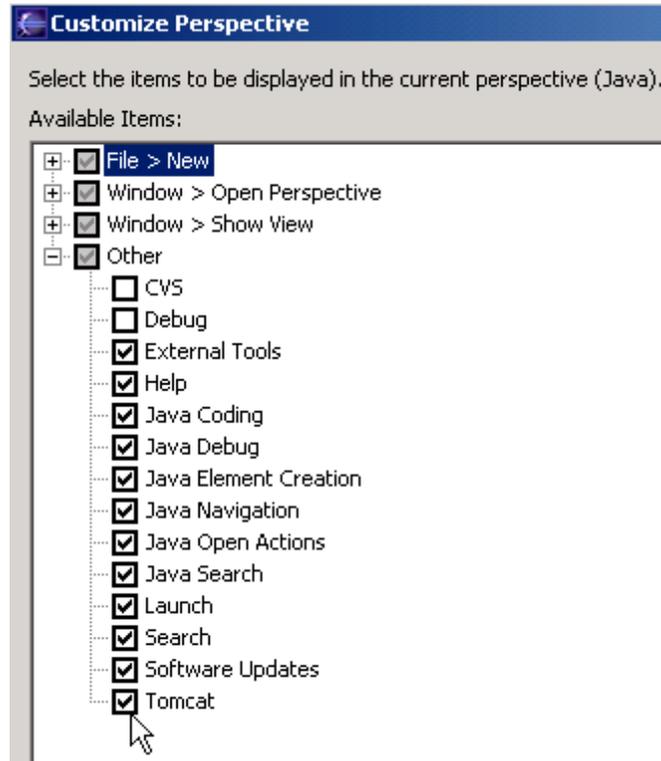


## Schritt 9: Menu Punkte für Tomcat aktivieren

In der Menuleiste auf *Window* 1k:

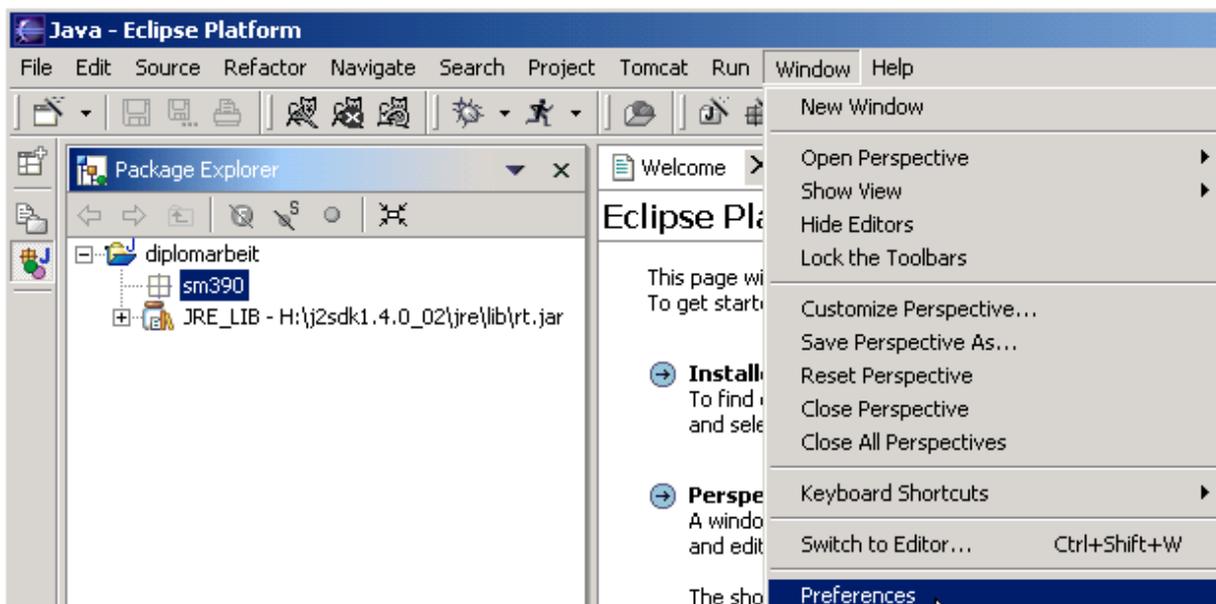


Anschließend Klick auf *Customize Perspektive*, es öffnet sich das *Customize Perspective*-Fenster. In diesem ist auf das kleine Plus vor *Other* zu klicken, um *Other* zu erweitern.

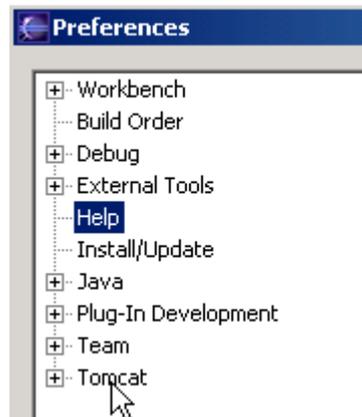


Setzen Sie ein Häkchen vor *Tomcat* und klicken Sie anschließend auf *OK*.

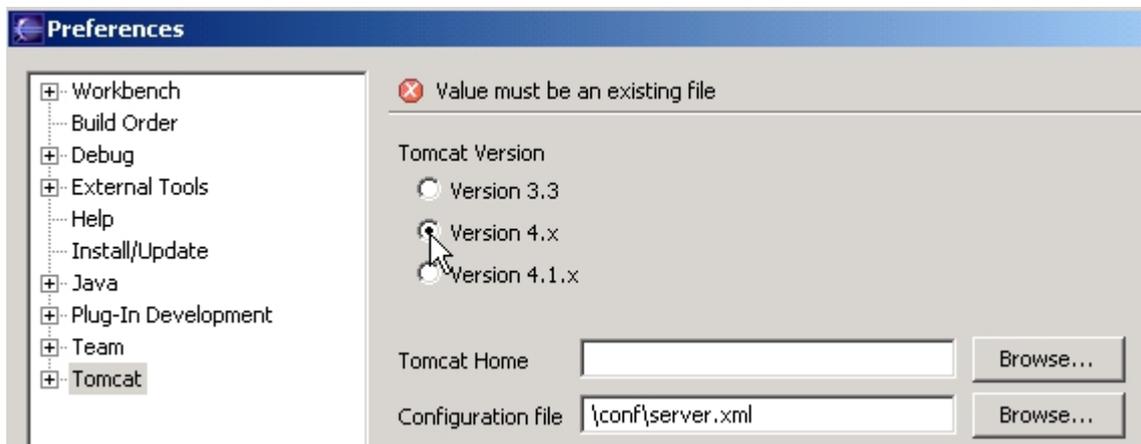
### Schritt 10: Plug-in Einstellungen überprüfen



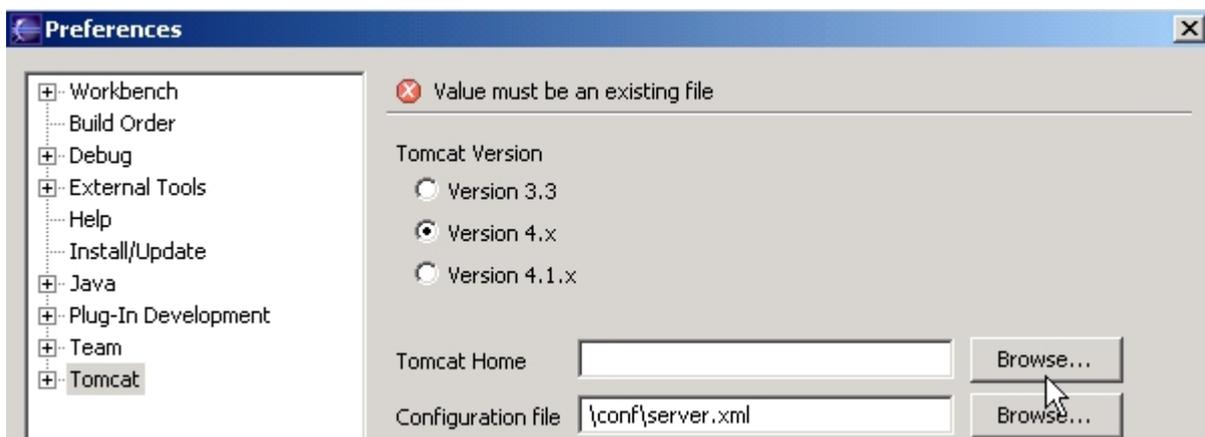
Ein Klick auf *Window*, anschließend auf *Preferences*.



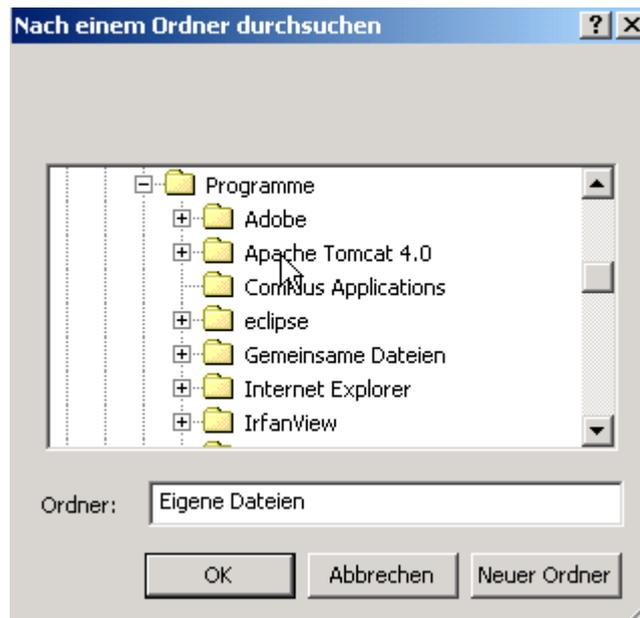
1k auf *Tomcat*.



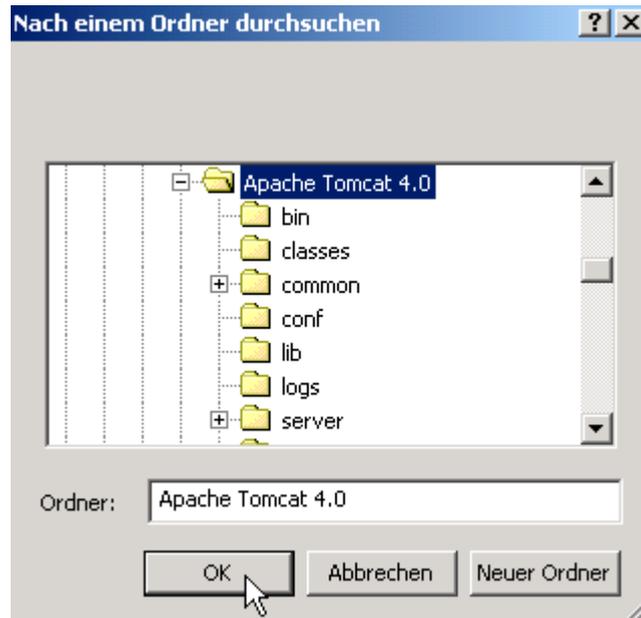
Version 4.x selektieren.



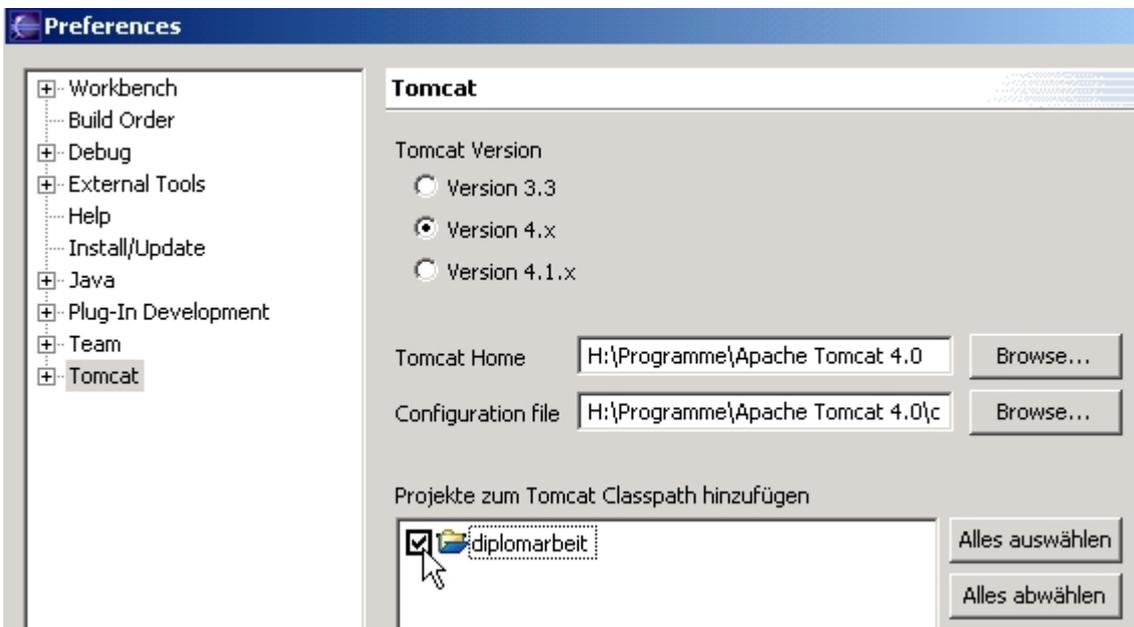
Klick auf *Browse*, rechts von *Tomcat Home*.



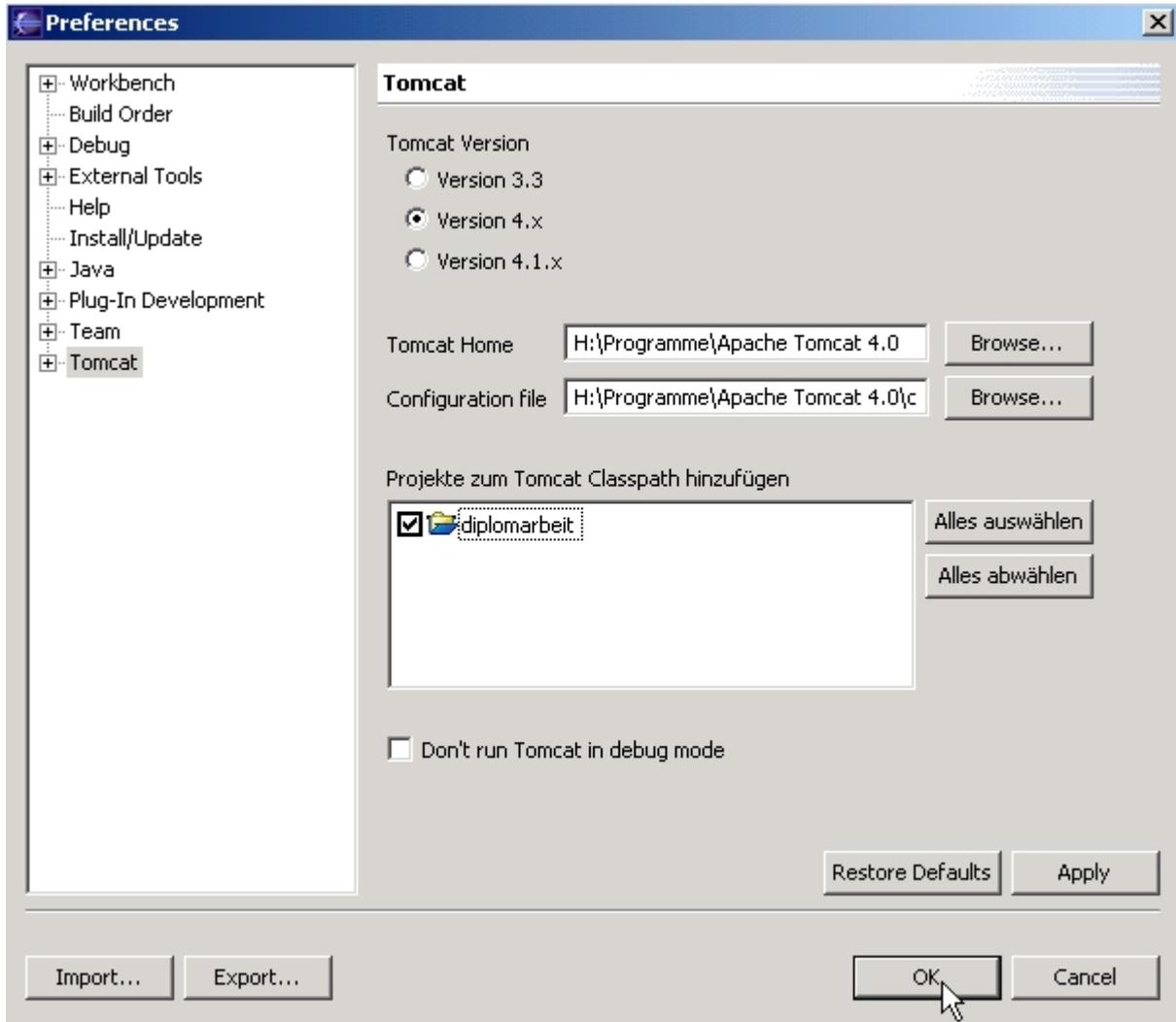
Markieren Sie durch Klick *H:\programme\Apache Tomcat 4.0*



Klicken Sie nun auf den Button *OK*.

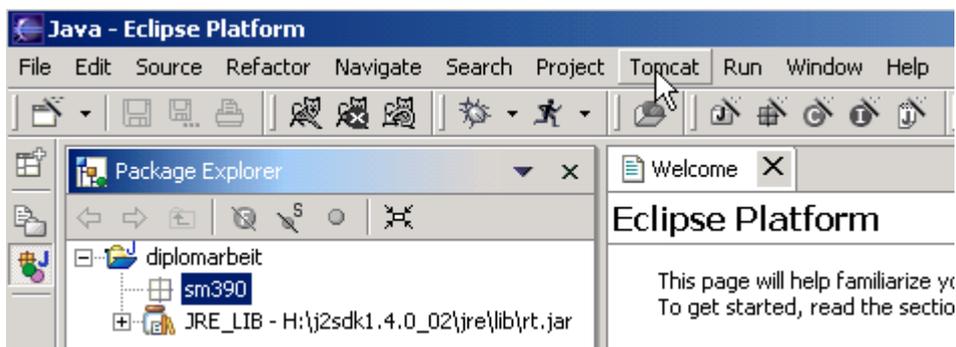


Setzen Sie ein Häkchen vor *Diplomarbeit*.

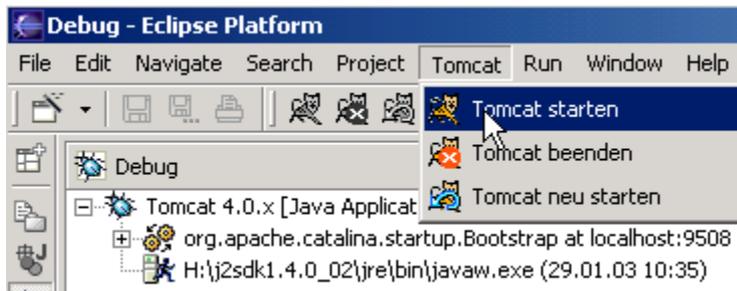


Klicken Sie abschließend auf **OK**.

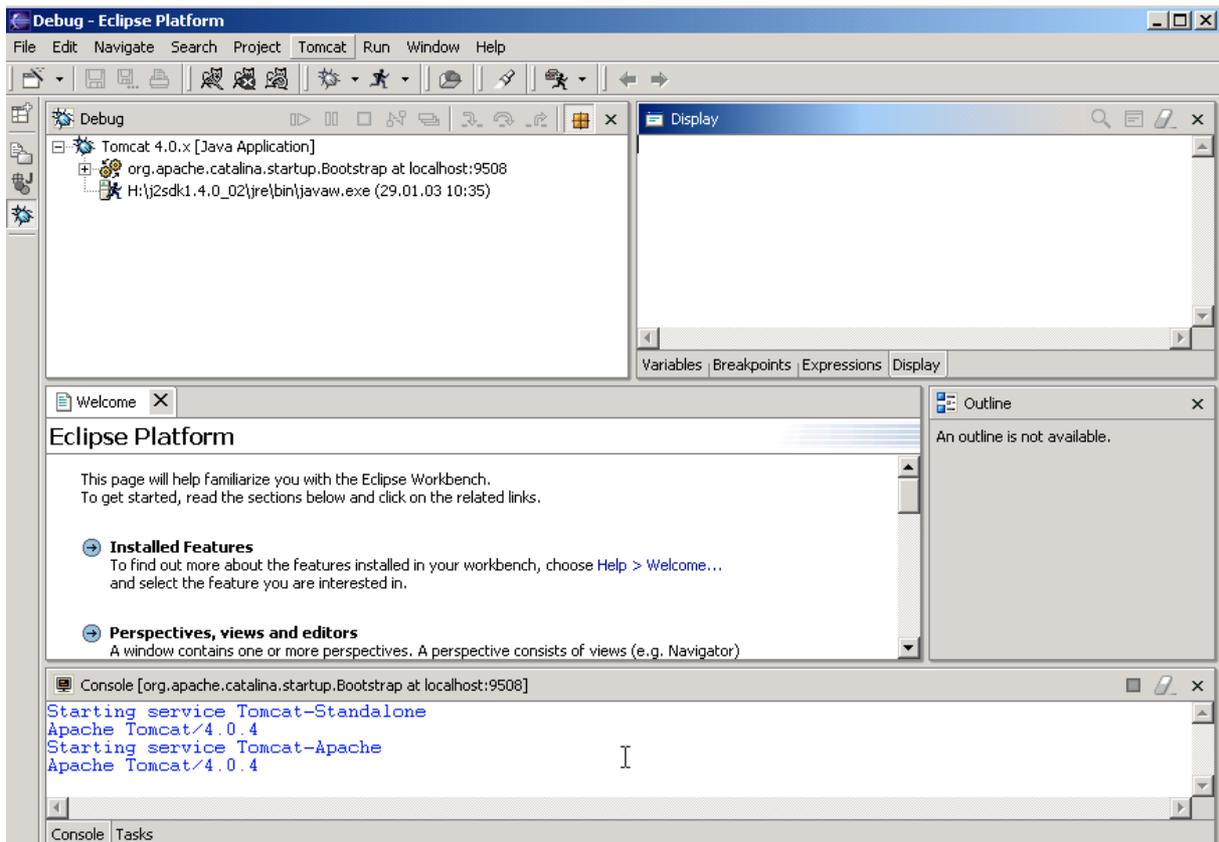
### Schritt 11: Tomcat starten und beenden



Klick auf *Tomcat*,



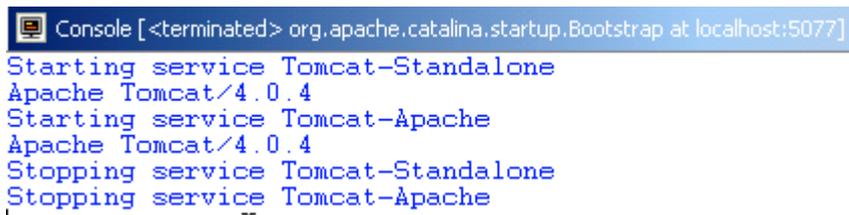
anschließend auf *Tomcat starten*.



Warten Sie einige Sekunden, bis die 4 blauen Zeilen erschienen sind. Dann ist der Start von Tomcat abgeschlossen.

Damit wurden Eclipse und Tomcat erfolgreich installiert.

Klick auf *Tomcat*, anschließend auf *Tomcat beenden*.

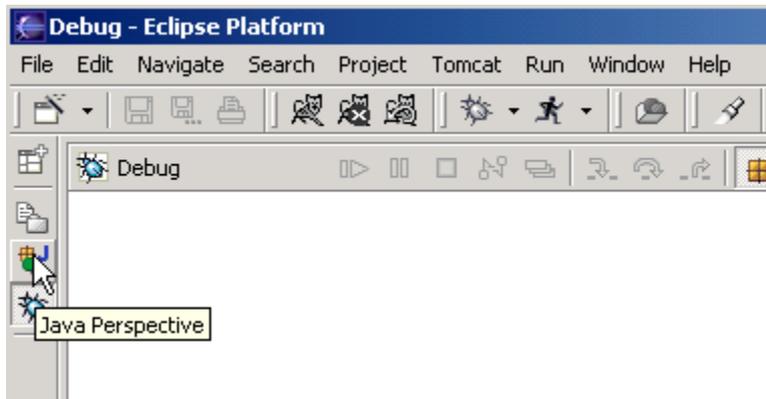


Nach einer kurzen Weile sind zwei weitere blaue Zeilen erschienen: "Stopping service Tomcat-Standalone" und "Stopping service Tomcat-Apache". Dann wurde Tomcat erfolgreich beendet.

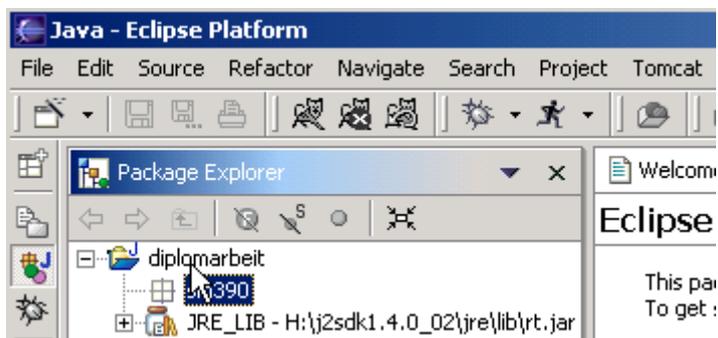
Schließen können Eclipse jetzt beenden (Klick auf *File* und anschließend auf *Exit*).

## 4. Eclipse konfigurieren

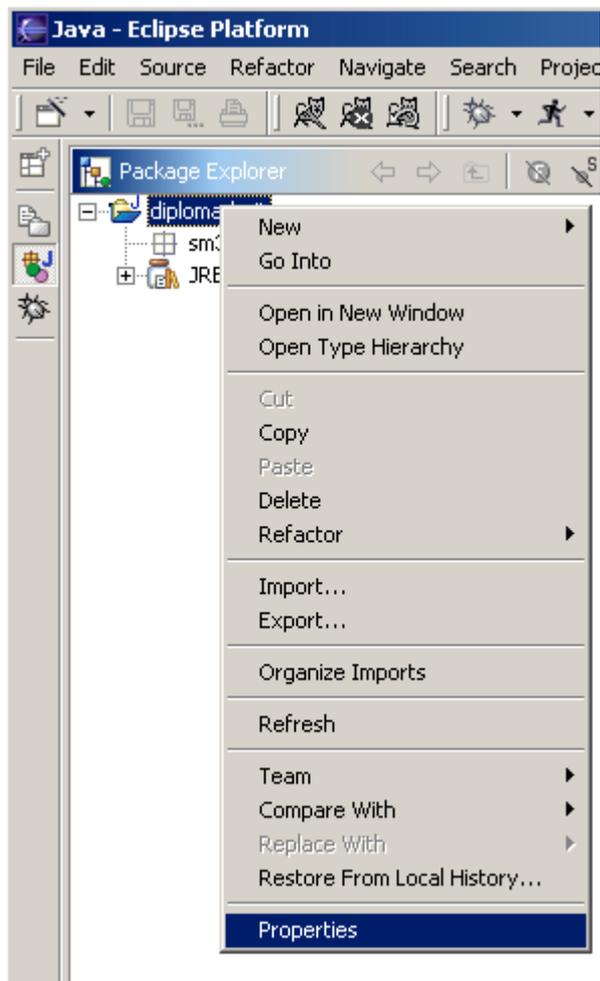
Starten Sie Eclipse.



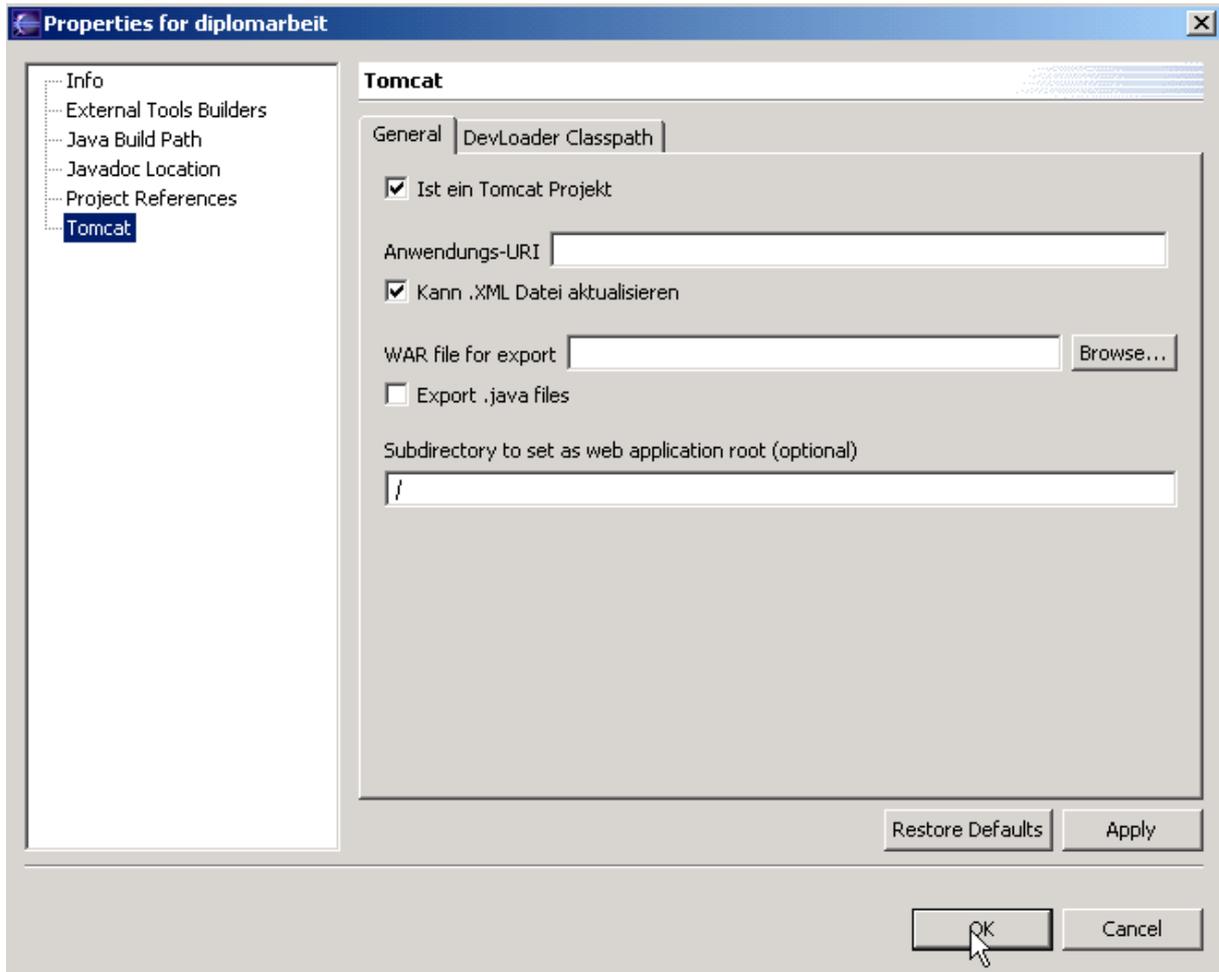
Ein Klick auf *Java Perspective*.



Klicken Sie mit der rechten Maustaste auf *diplomarbeit*.

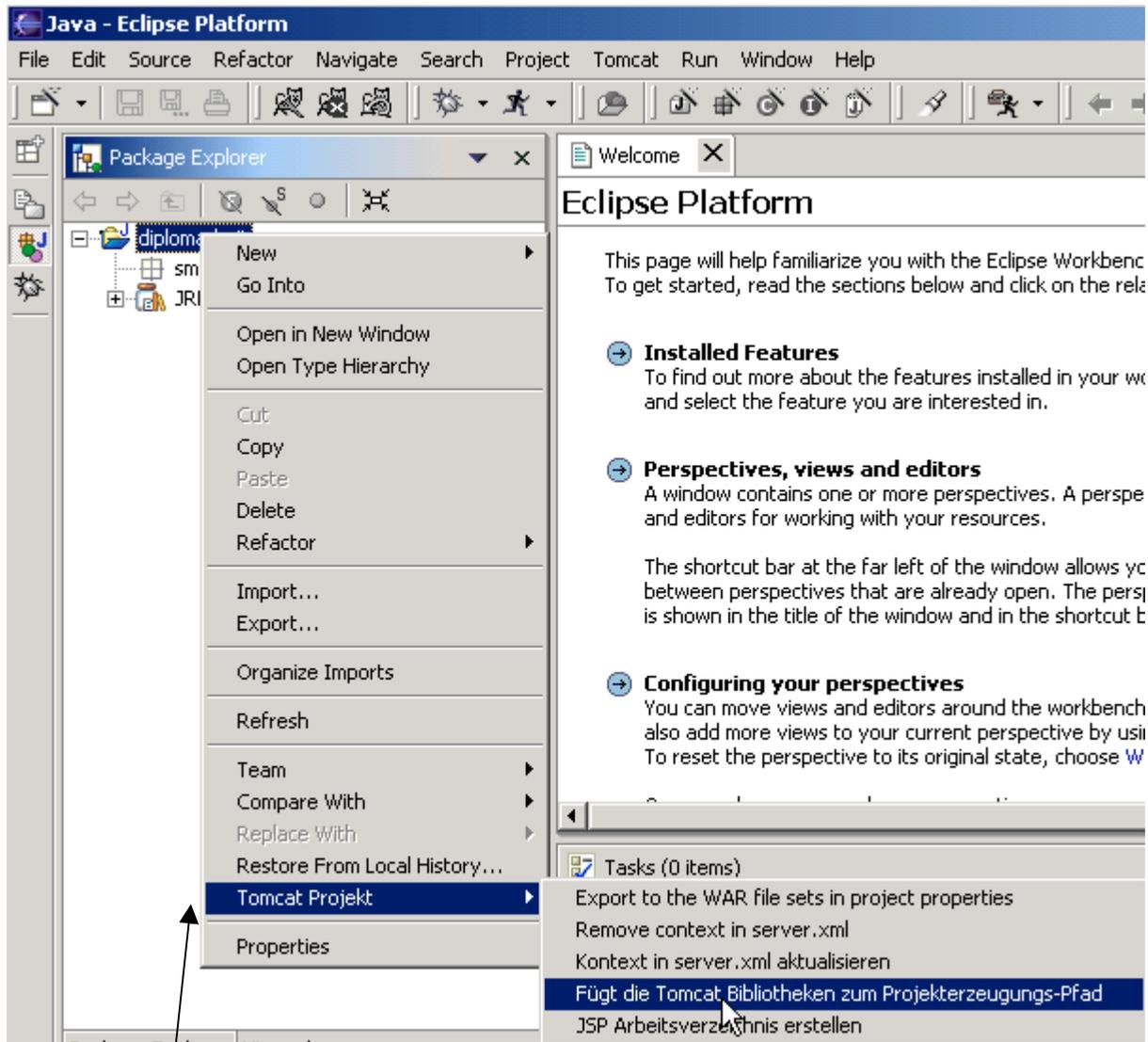


Anschließend auf *Properties* klicken.

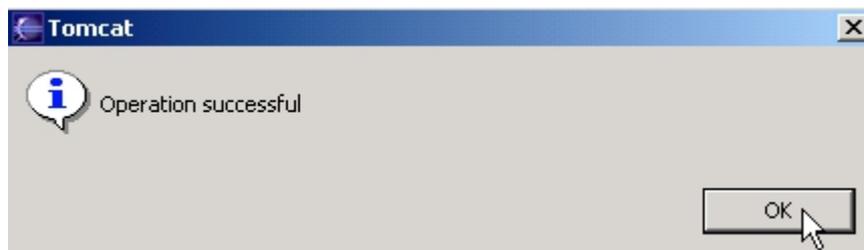


Klicken Sie auf *Tomcat* und stellen Sie alle Optionen, wie auf dem oben dargestellten Bild ein. Insbesondere muß ein Häkchen vor *Ist ein Tomcat Projekt* und eins vor *Kann .XML Datei aktualisieren* sein.

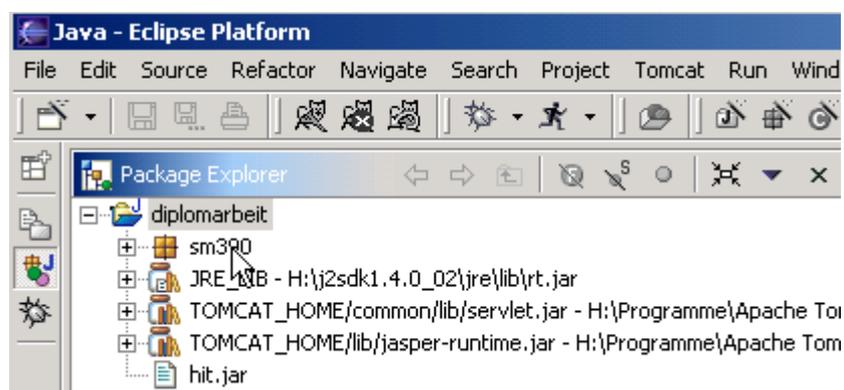
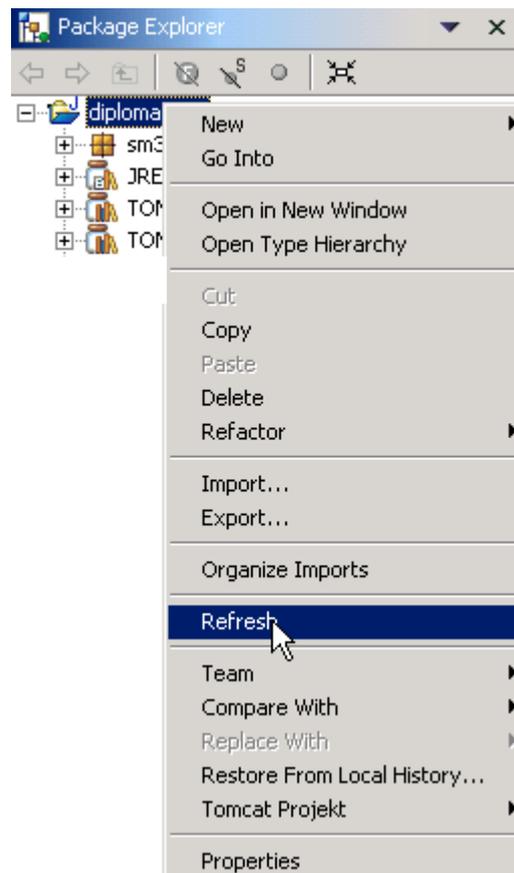
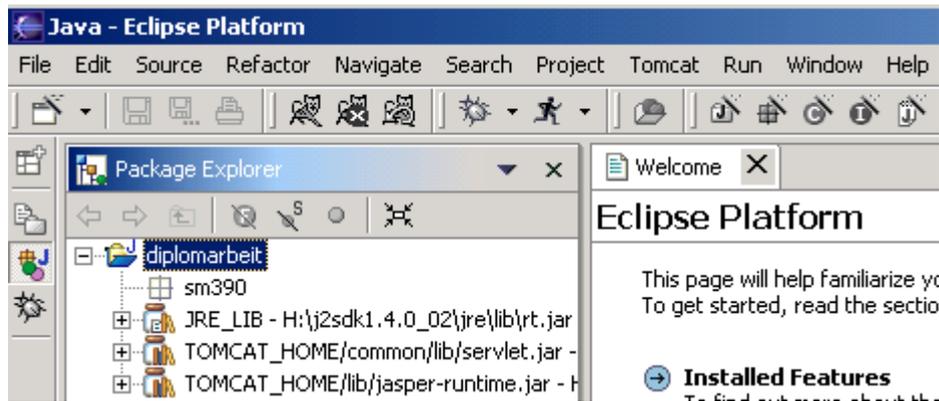
Abschließend auf *OK* klicken.



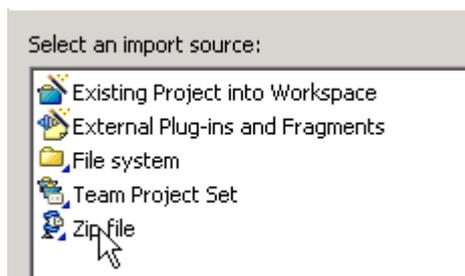
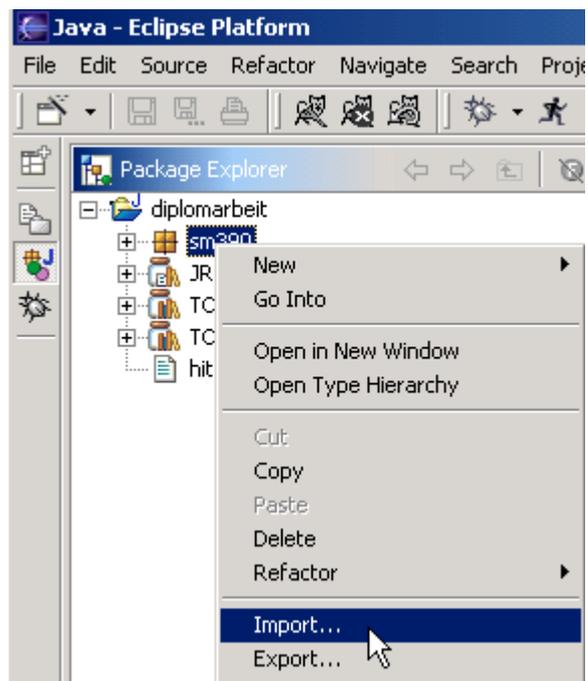
Der Eintrag Tomcat Projekt war vorher nicht vorhanden.



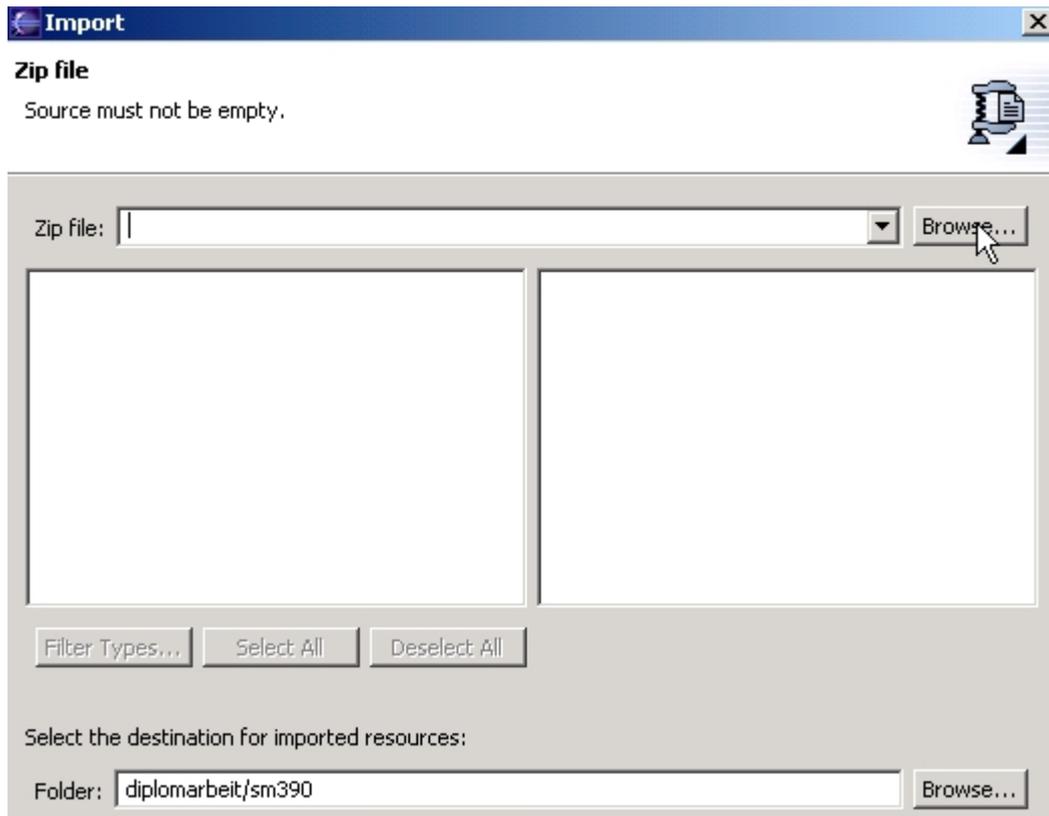
Mit rechter Maustaste Klick auf *diplomarbeit* → *Tomcat Projekt*  
 → *Fügt die Tomcat Bibliotheken zum Projekterzeugungs-Pfad* → Ausgabe *Operation successful*



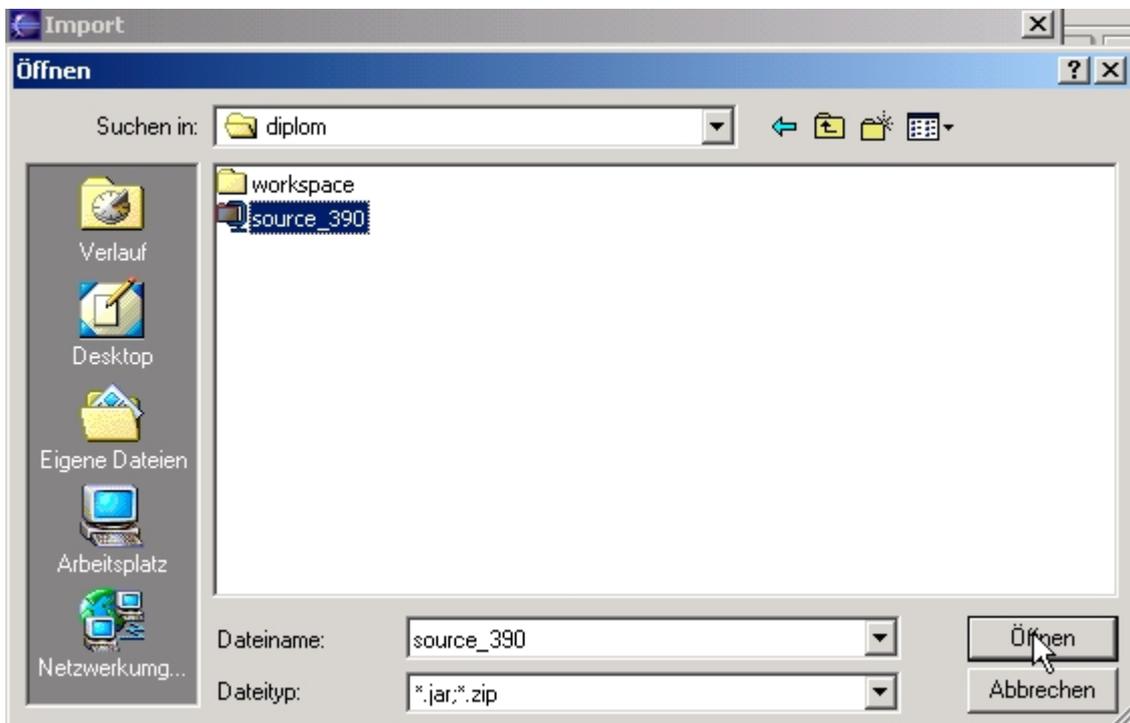
Mit rechter Maustaste Klick auf *diplomarbeit* → *Refresh*  
 → *hit.jar* wird vom Package Explorer angezeigt



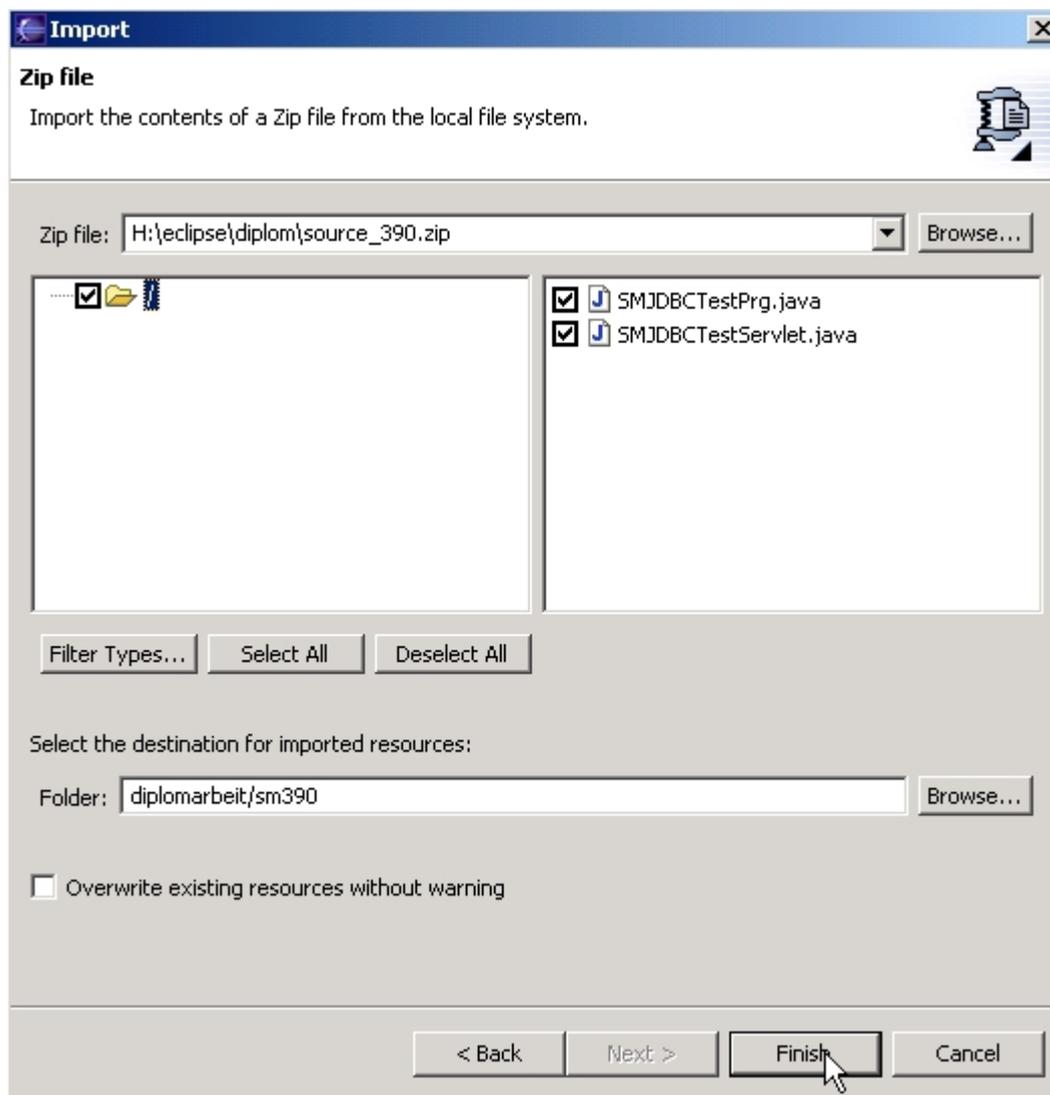
Mit rechter Maustaste auf *sm390* klicken → *Import... 1k* → *Zip file 1k* → *Next 1k*



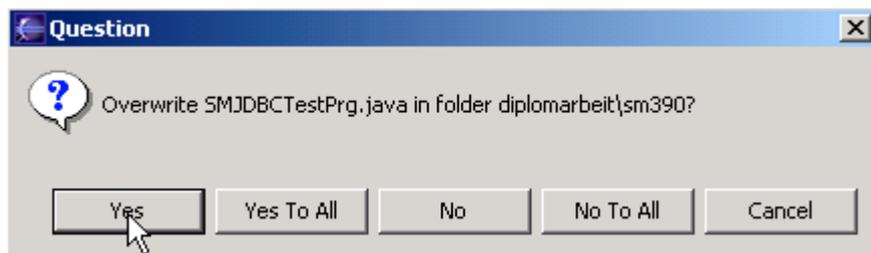
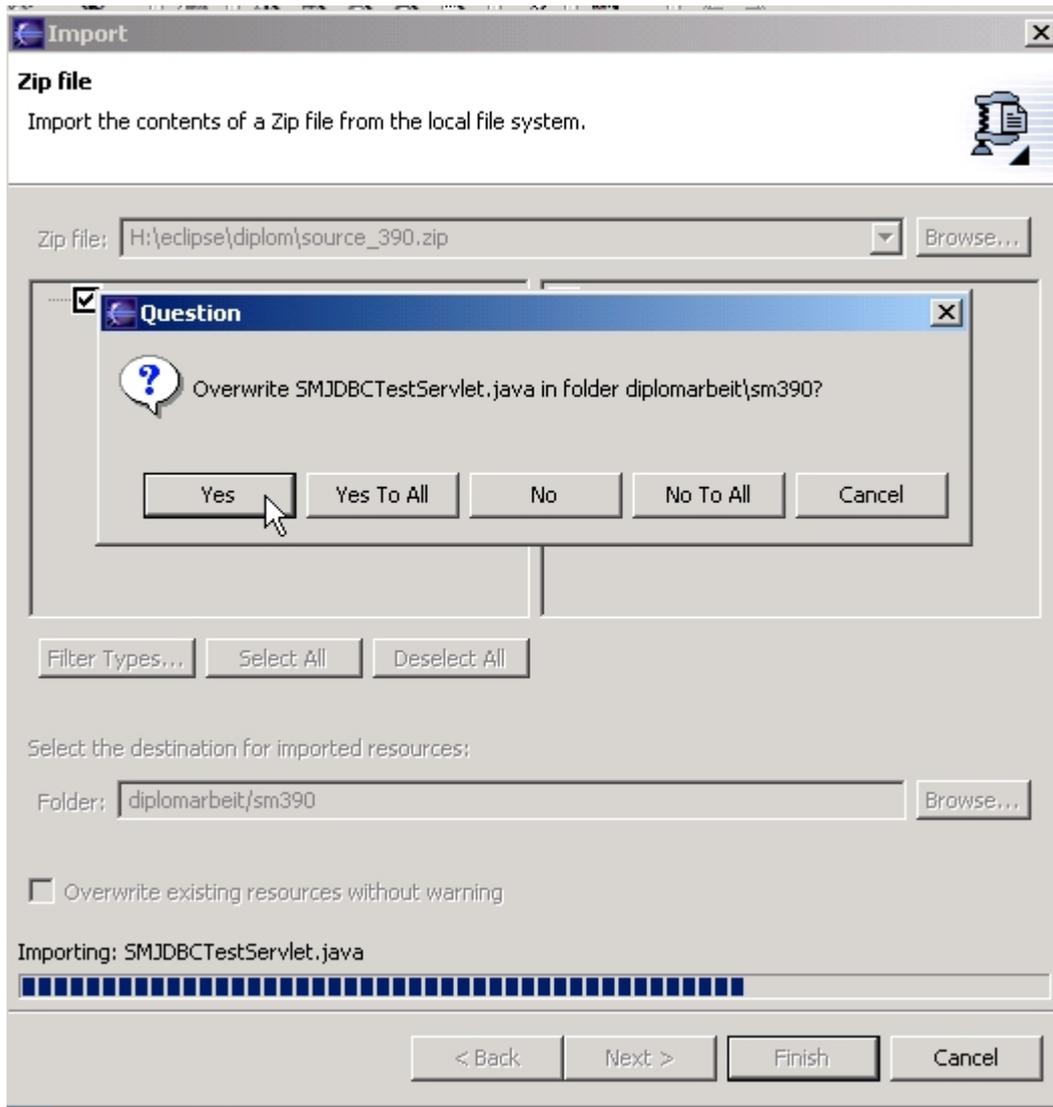
Klick auf *Browse* rechts neben *Zip file* und nach *H:\Eclipse\diplom* gehen



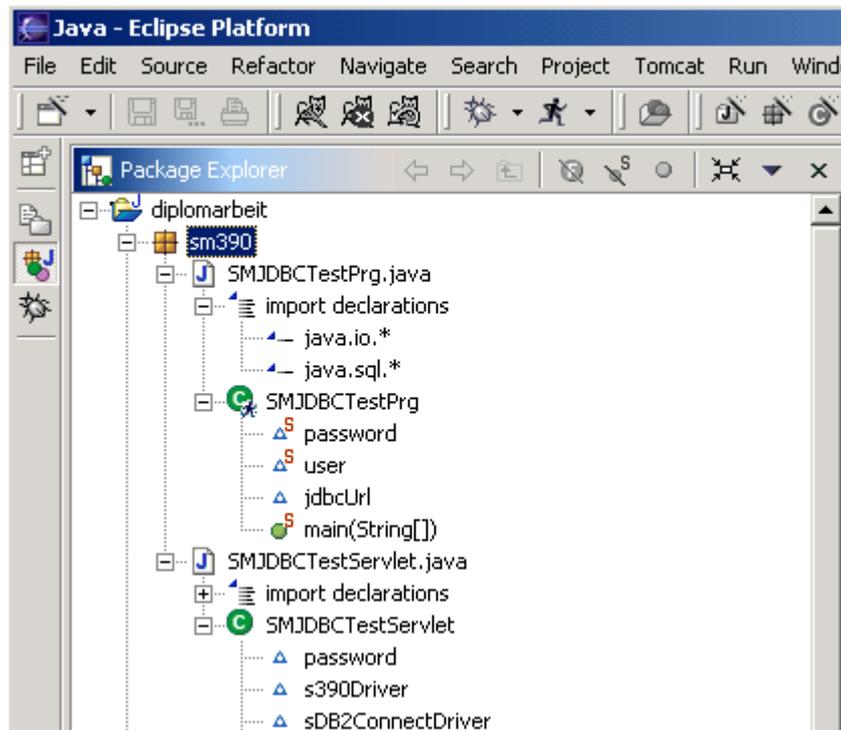
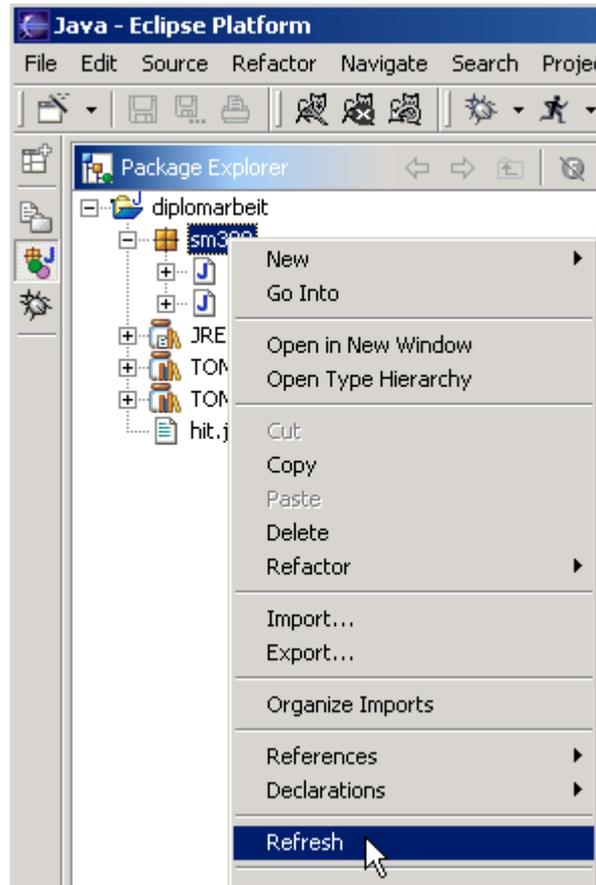
*source\_390* markieren 1k, öffnen 1k.



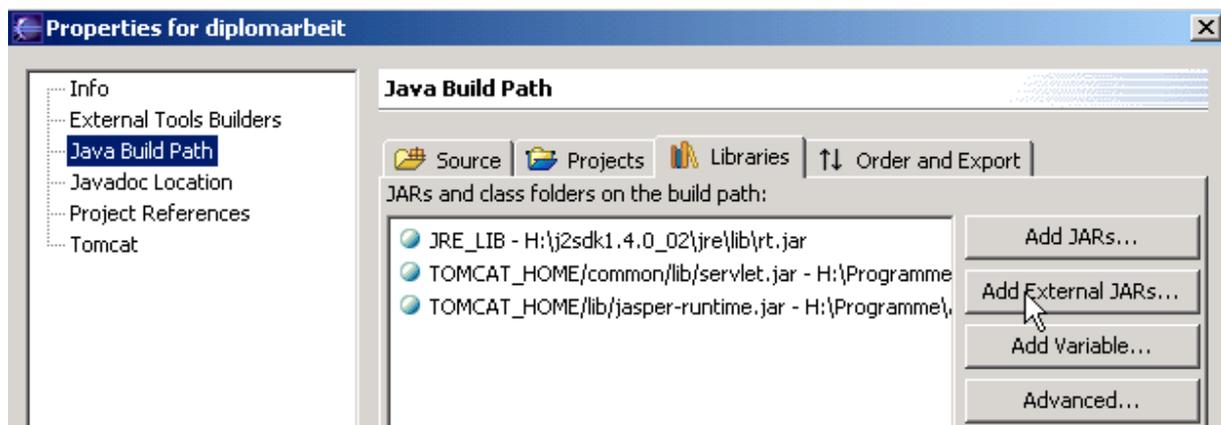
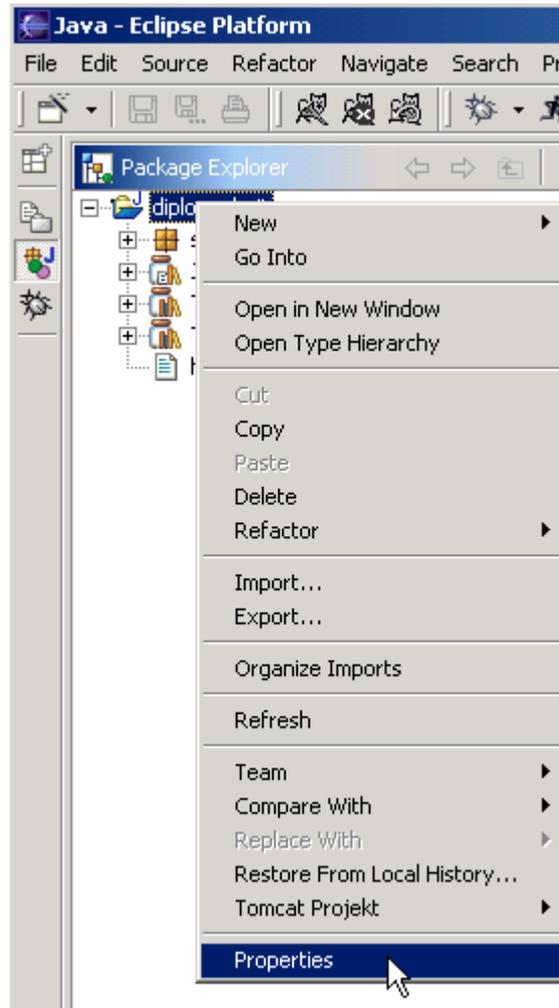
Klick auf den Button *Finish*.



Die beiden Fragen nach dem Überschreiben mit Yes beantworten, Fenster verschwindet.

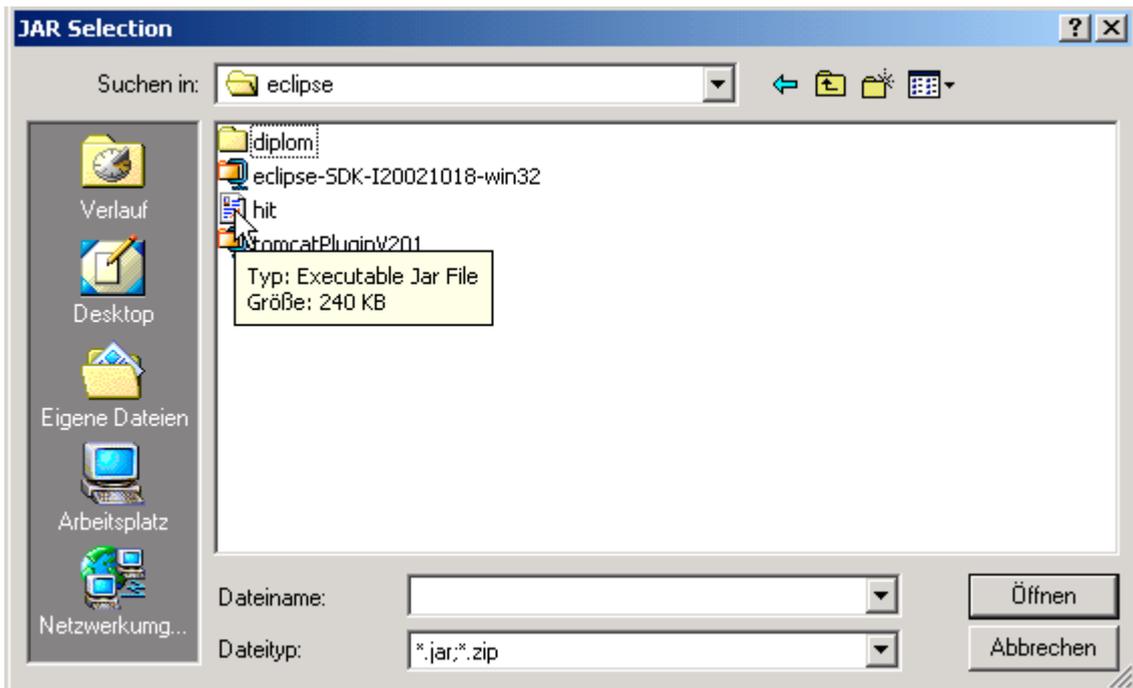


Jetzt sieht man die Klassen.

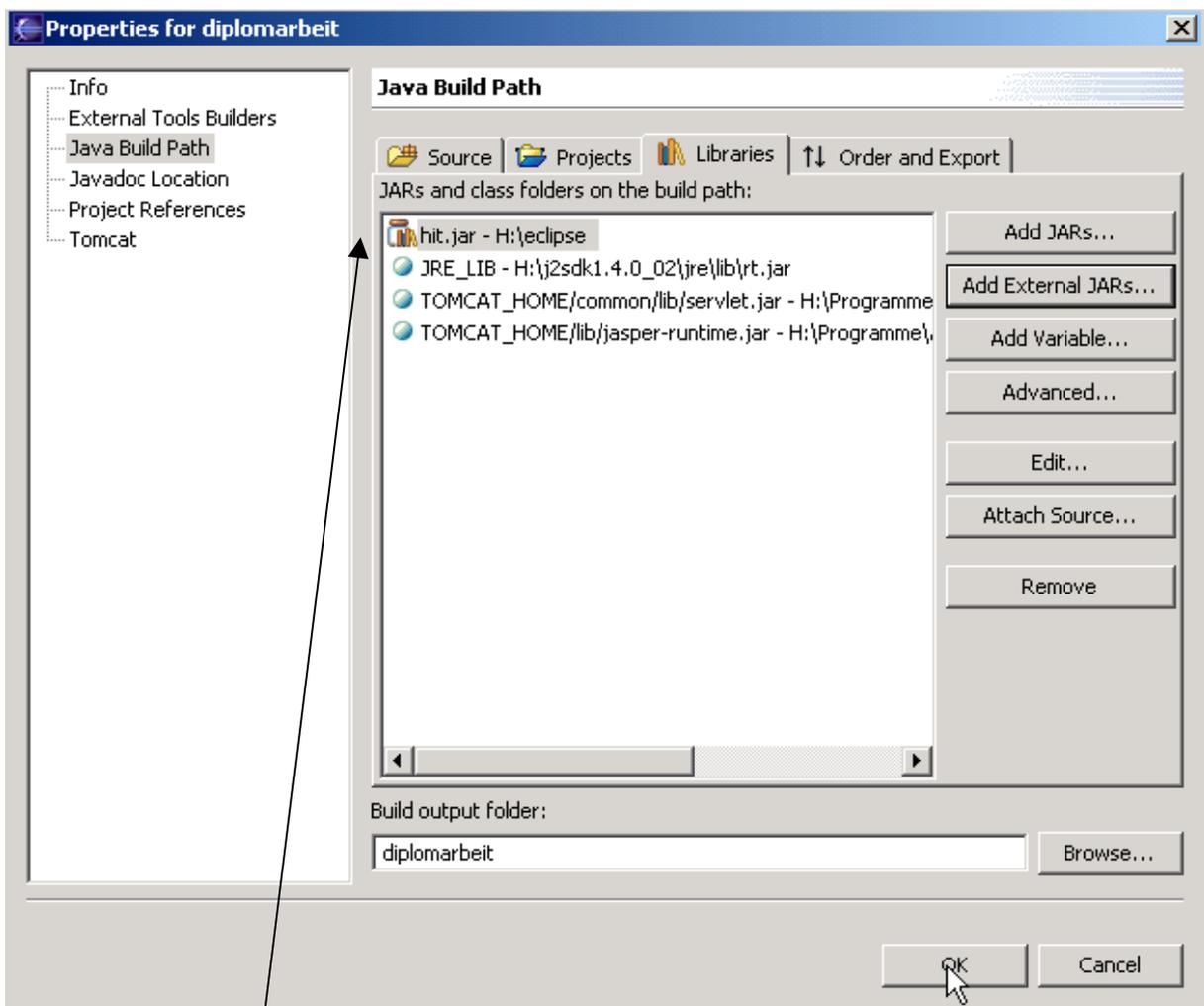


Auf diplomarbeit 1kr, auf Properties 1k

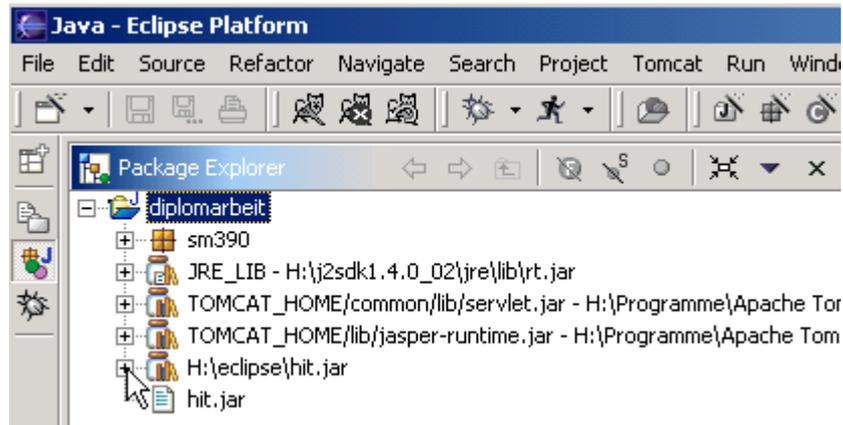
Auf Java Build path 1k, 1k auf Registerkarte Libraries, 1k auf Add external Jars



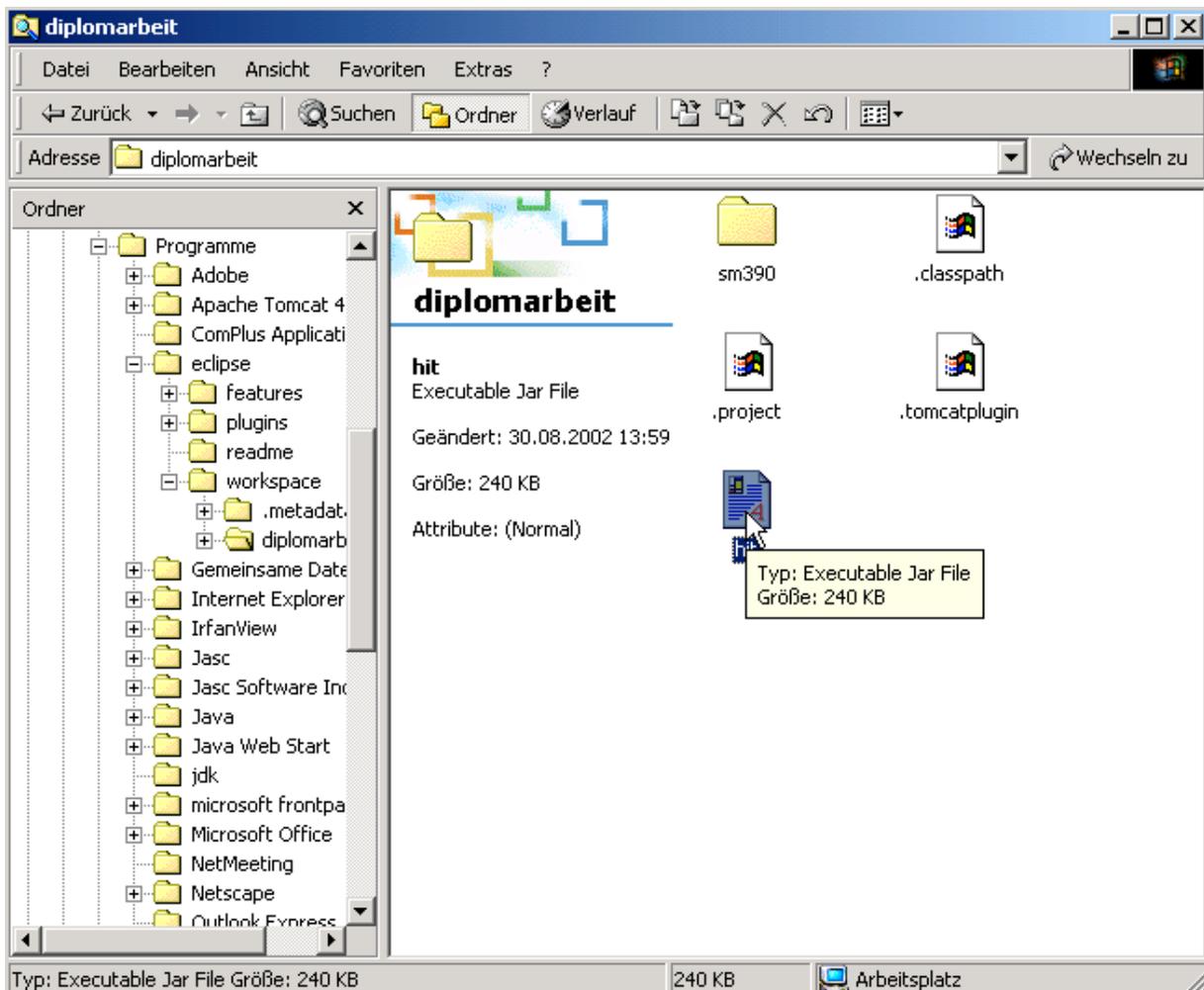
nach *H:\Eclipse* gehen, 1k auf *hit*, *Öffnen* 1k



Jetzt ist *hit.jar* dabei, 1k auf *OK*

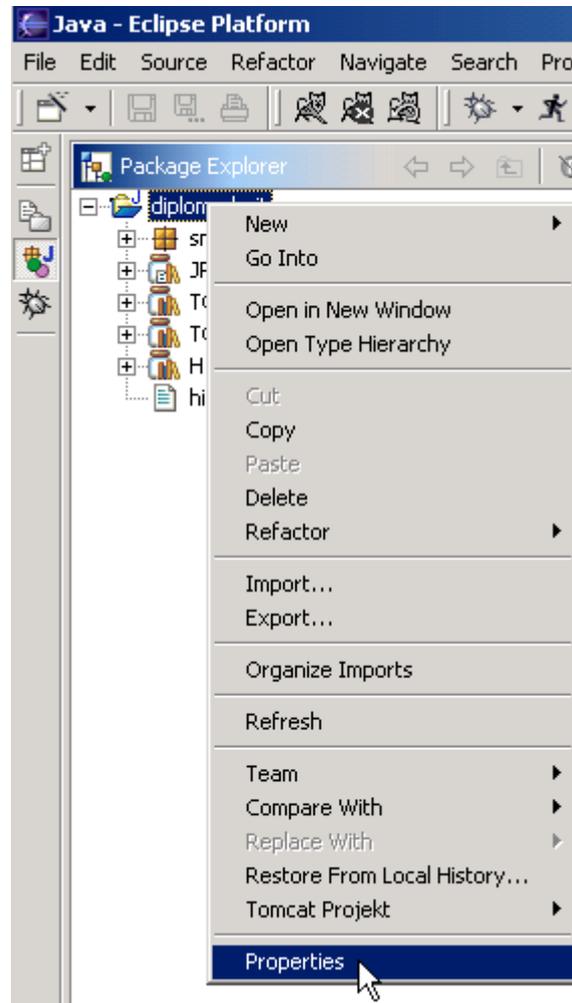


Jetzt ist *hit.jar* neu dazugekommen.

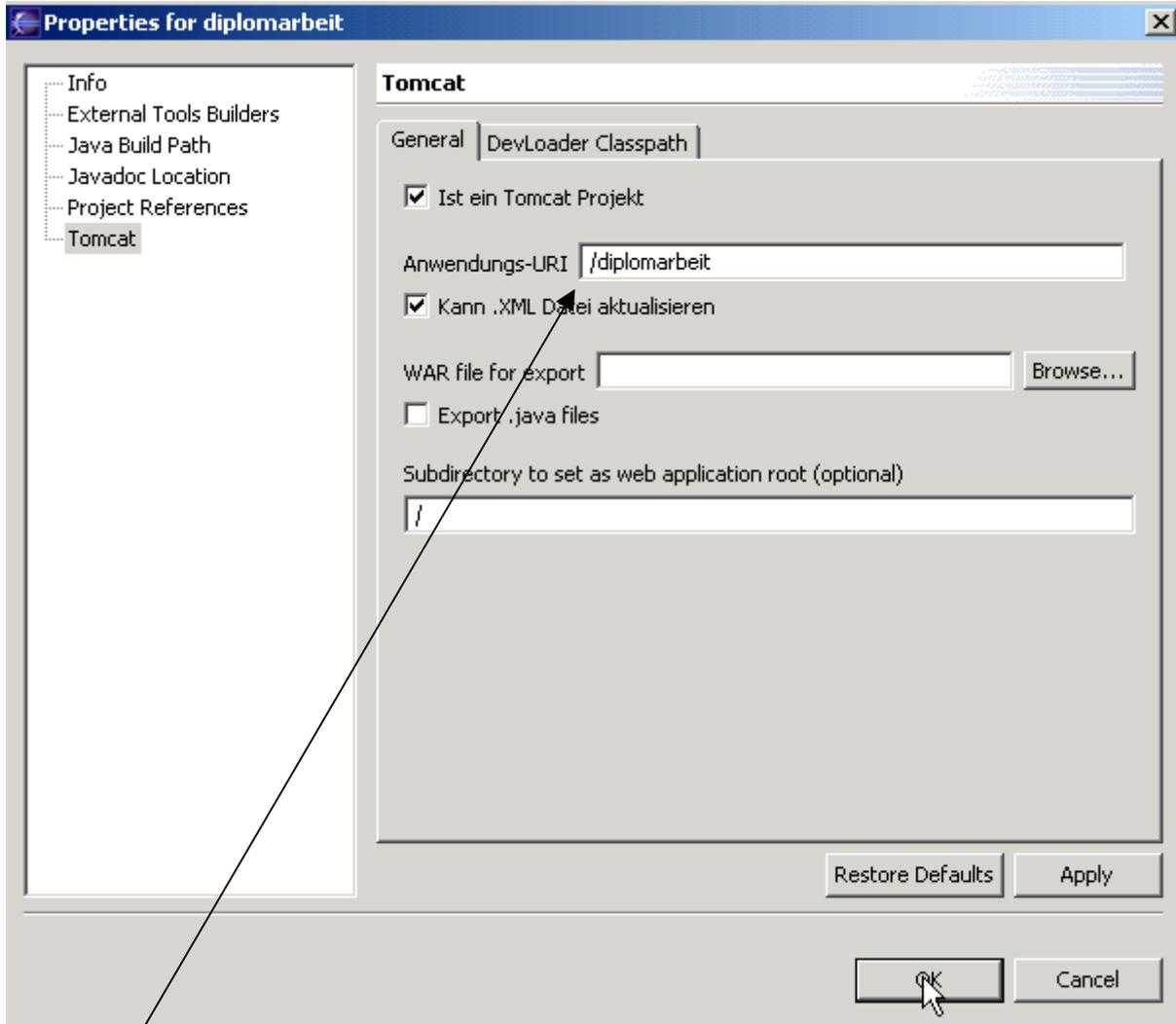


Im Windows explorer *h:\programme\eclipse\workspace\diplomarbeit* öffnen

Doppelklick auf *hit.jar*, Windows Explorer schliessen



1kr auf *Diplomarbeit*, *Properties* 1k, 1k auf *Tomcat*

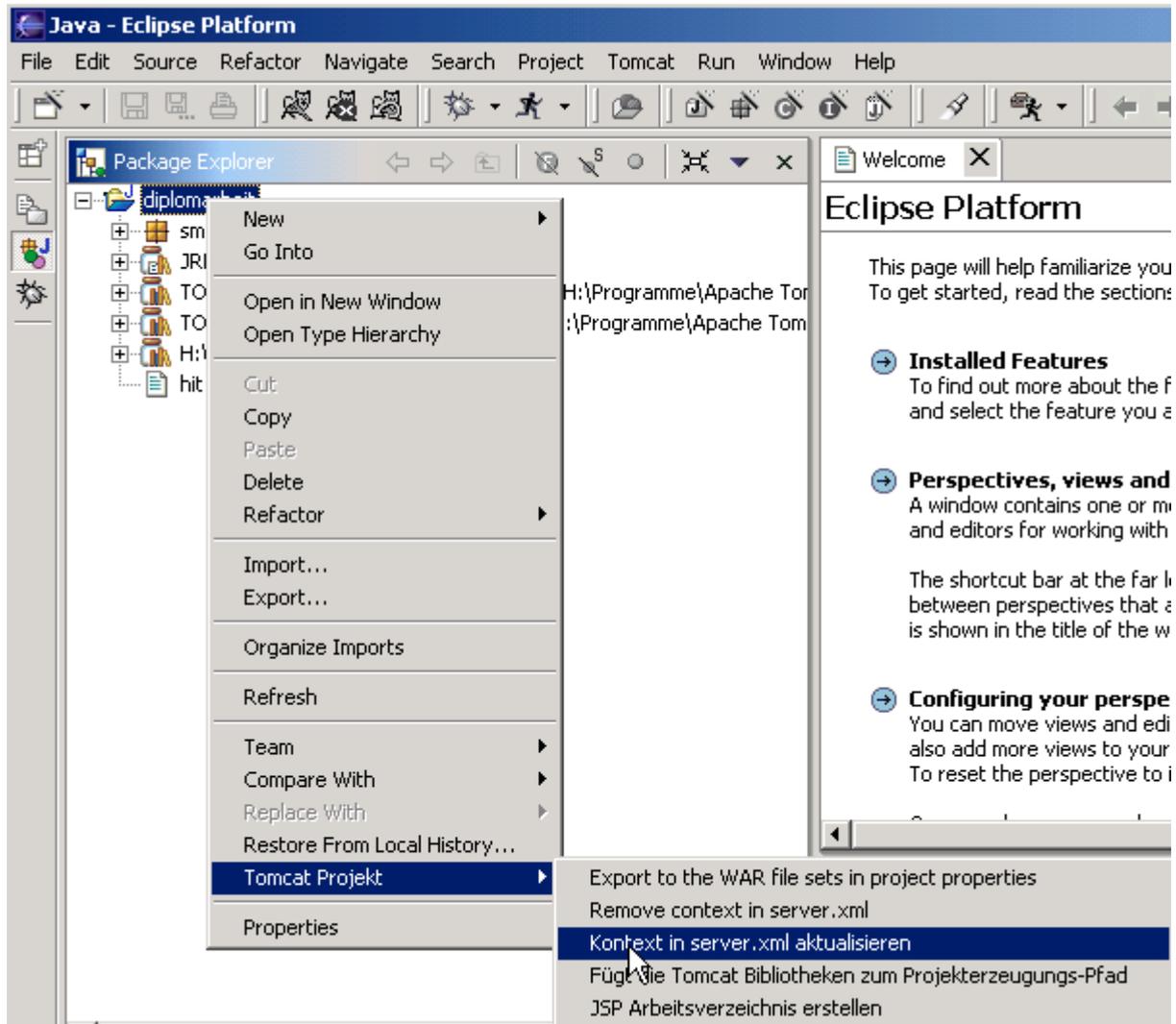


/diplomarbeit einfügen,

Häkchen müssen vor *Ist ein Tomcat Projekt* sowie *kann .XML Dateien aktualisieren* stehen.

1k auf OK

## 5. In Betrieb nehmen



*diplomarbeit 1kr, tomcat projekt auswählen, Kontext in Server .XML aktualisieren 1k*



Klick auf OK, und das *Tomcat*-Fenster wird geschlossen.

Klick auf *Tomcat*, anschließend auf *Tomcat starten*.

Warten, bis die vier blauen Zeilen

*Starting service Tomcat-Standalone*  
*Apache Tomcat/4.0.4*  
*Starting service Tomcat-Apache*  
*Apache Tomcat/4.0.4*

angezeigt werden.

Öffnen Sie einen Browser, z.B. Netscape oder den Internet Explorer.

Dort als Internet-Adresse (beim deutschsprachigen Explorer die Zeile *Adresse*, beim englischsprachigen Netscape die Zeile *Location*)

die URL *http://localhost:8080/servlet/sm390.SMJDBCtestServlet* eingeben.

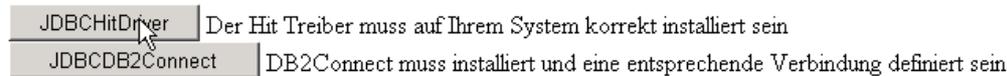
Es erscheint das *Hello World Servlet*.



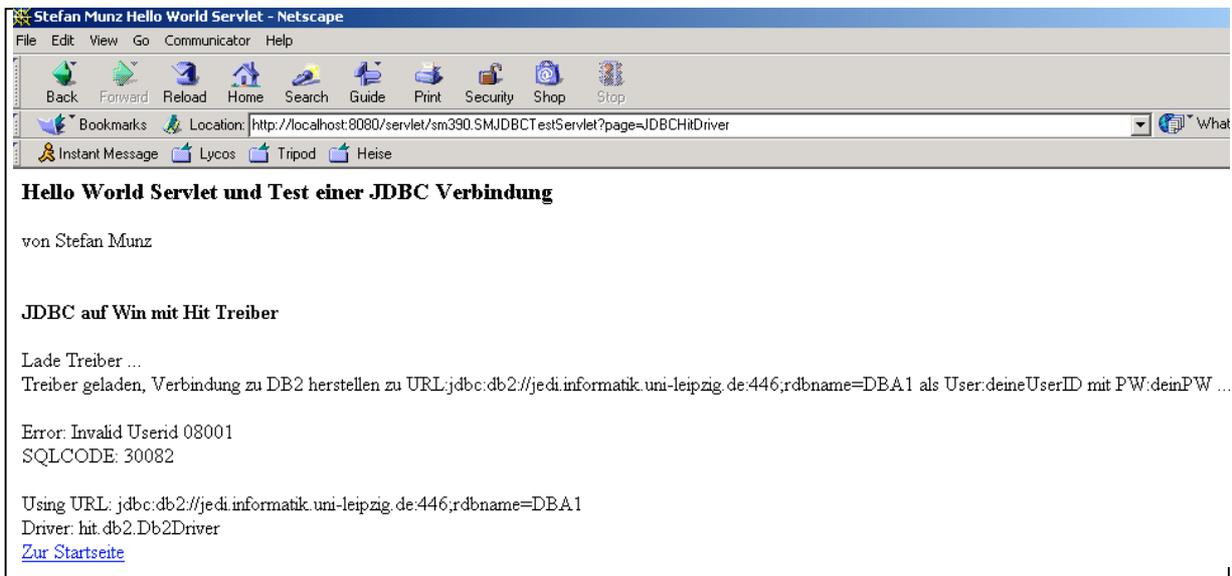
## Hello World Servlet und Test einer JDBC Verbindung

von Stefan Munz

Durch Klick auf einen der unteren Buttons wird ein JDBC Zugriff auf die DB2 Datenbank auf der Leipziger 390 durchgeführt. Je nachdem auf welchem Betriebssystem das Servlet läuft, erhalten Sie eine oder 2 Optionen für den DB-Zugriff:

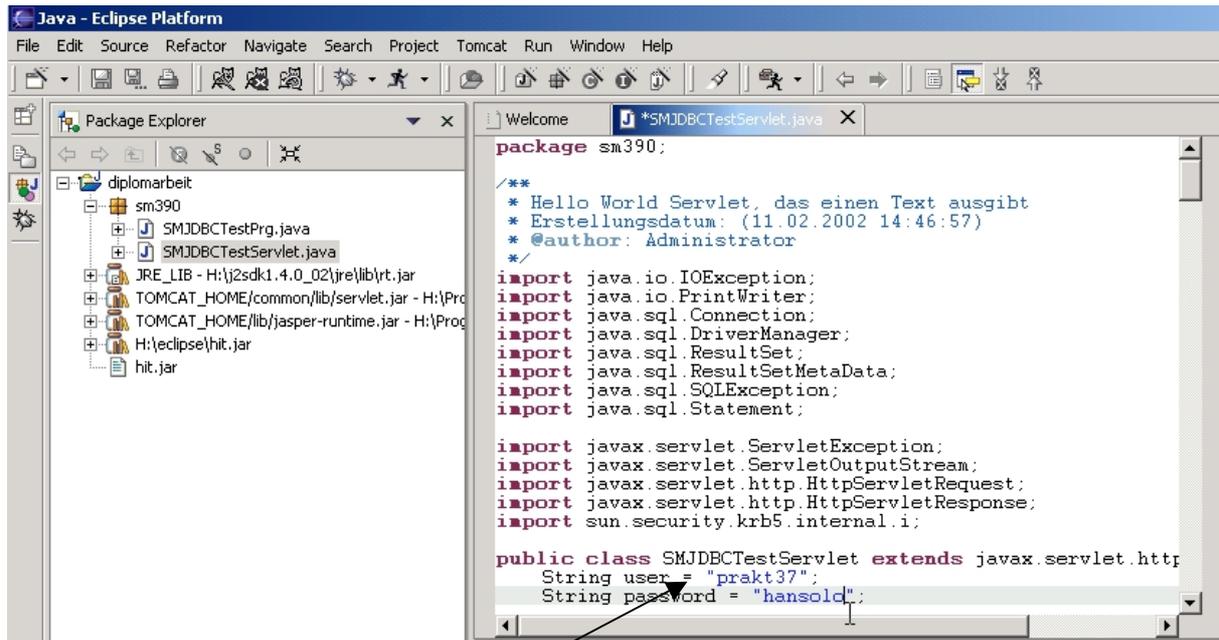


Klick auf *JDBCHitDriver*.



Es wird versucht, auf die Datenbank *DBA1* des OS/390-Servers *je di.informatik.uni-leipzig* zuzugreifen. Nun ist das aber nur möglich, wenn über ein gültiges Login einschließlich dem korrekten Passwort zugegriffen wird. Nun kennt das Servlet aber weder Ihr Login auf dem Server, noch Ihr Passwort. Deshalb kommt obige Fehlermeldung.

Klick auf *Zur Startseite*.



*Aufgabe:* Tragen Sie hier ihre User-ID sowie ihr Passwort ein und speichern Sie diese Änderung..

Stefan Munz Hello World Servlet - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Guide Print Security Shop Stop

Bookmarks Location: <http://localhost:8080/servlet/sm390.SMJDBCtestServlet>

Instant Message Lycos Tripod Heise

### Hello World Servlet und Test einer JDBC Verbindung

von Stefan Munz

Durch Klick auf einen der unteren Buttons wird ein JDBC Zugriff auf die DB2 Datenbank auf der Leipziger 390 durchgeführt. Je nachdem auf welchem Betriebssystem das Servlet läuft, erhalten Sie eine oder 2 Optionen für den DB-Zugriff:

JDBCHitDriver	Der Hit Treiber muss auf Ihrem System korrekt installiert sein
JDBCDB2Connect	DB2Connect muss installiert und eine entsprechende Verbindung definiert sein

Jetzt erneut 1k auf *JDBCHitDriver*.

## Hello World Servlet und Test einer JDBC Verbindung

von Stefan Munz

### JDBC auf Win mit Hit Treiber

Lade Treiber ...

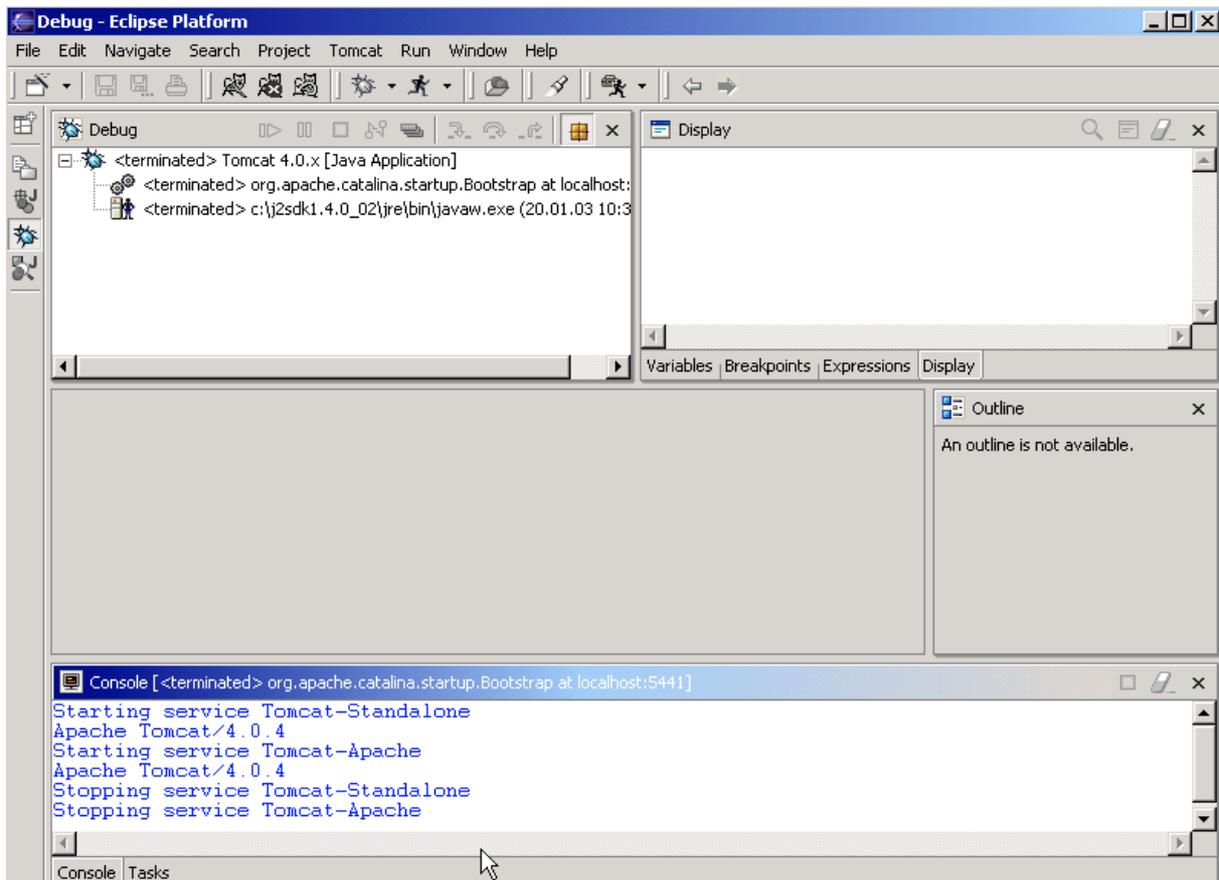
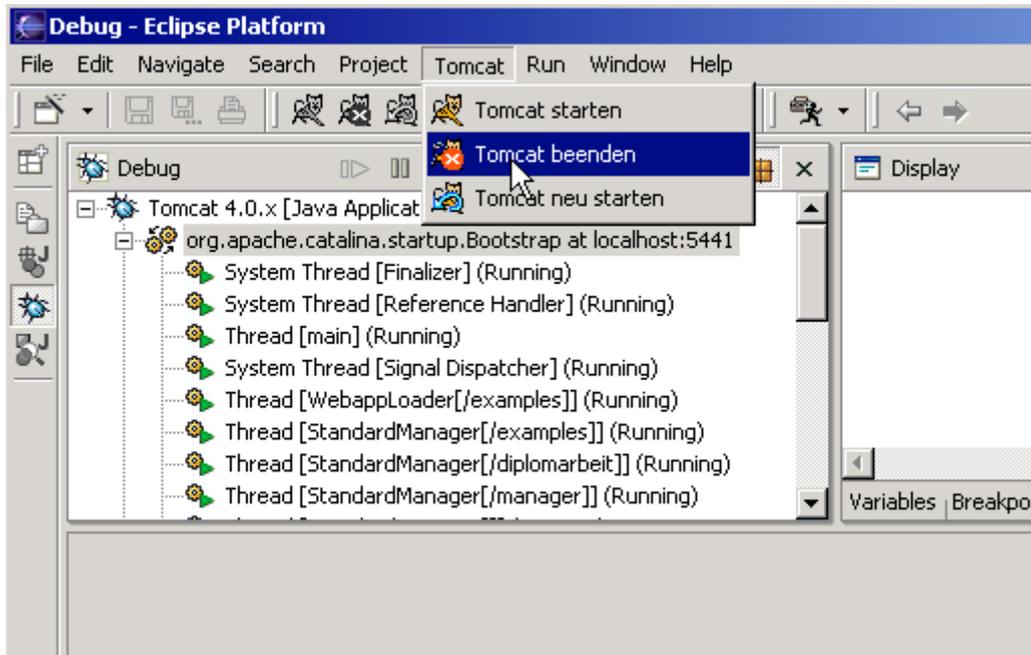
Treiber geladen, Verbindung zu DB2 herstellen zu URL:jdbc:db2://jedi.informatik.uni-leipzig.de:446;rdname=DBA1 als User:prakt51 mit PW:db2 .  
Verbindung hergestellt. Führe SQL Statement aus...

NAME	CREATOR	TYPE	DBNAME	TSNAME
SYSCOPY	SYSIBM	T	DSNDB06	SYSCOPY
SYSCOLAUTH	SYSIBM	T	DSNDB06	SYSDBASE
SYSCOLUMNS	SYSIBM	T	DSNDB06	SYSDBASE
SYSFOREIGNKEYS	SYSIBM	T	DSNDB06	SYSDBASE
SYSINDEXES	SYSIBM	T	DSNDB06	SYSDBASE
SYSINDEXPART	SYSIBM	T	DSNDB06	SYSDBASE
SYSKEYS	SYSIBM	T	DSNDB06	SYSDBASE
SYSLINKS	SYSIBM	T	DSNDB06	SYSDBASE
SYSRELS	SYSIBM	T	DSNDB06	SYSDBASE
SYSSYNONYMS	SYSIBM	T	DSNDB06	SYSDBASE
SYSTABAUTH	SYSIBM	T	DSNDB06	SYSDBASE
SYSTABLEPART	SYSIBM	T	DSNDB06	SYSDBASE
SYSTABLES	SYSIBM	T	DSNDB06	SYSDBASE
SYSTABLESPACE	SYSIBM	T	DSNDB06	SYSDBASE

Nun wird die DB2-Tabelle vom S/390-Server korrekt übertragen.

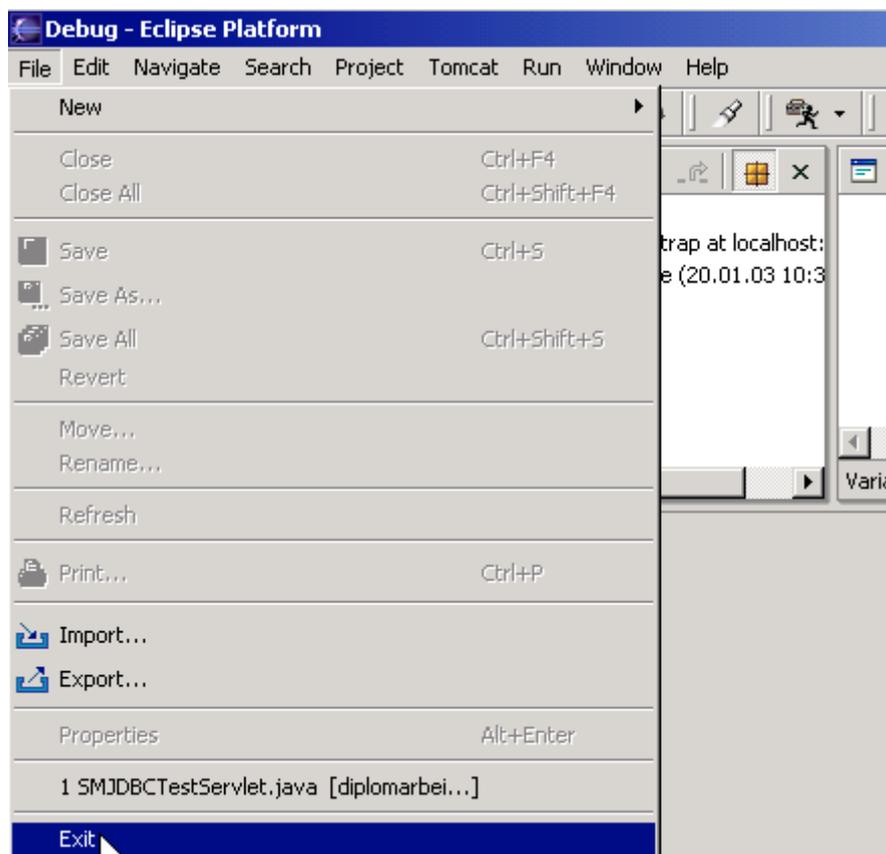
**Aufgabe:** Ersetzen Sie im Servlet "von Stefan Munz" durch "durchgeführt von <ihr Name>". Führen Sie dann das Servlet erneut aus. Erzeugen Sie einen Screenshot entsprechend der obigen Abbildung, der jetzt ihren Namen enthält und schicken Sie diesen per E-Mail an ihren Betreuer. Dieser darf max. 250 KByte groß sein. Verwenden Sie möglichst das JPEG-Format, so läßt sich der Screenshot auf weniger als 90 KByte komprimieren.

## 6. Herunterfahren



Klicken Sie auf *Tomcat*, anschließend auf *Tomcat beenden*.

Warten Sie, bis die beiden blauen Zeilen "Stopping service ..." im Fenster der Java-Eclipse Platform angezeigt werden.



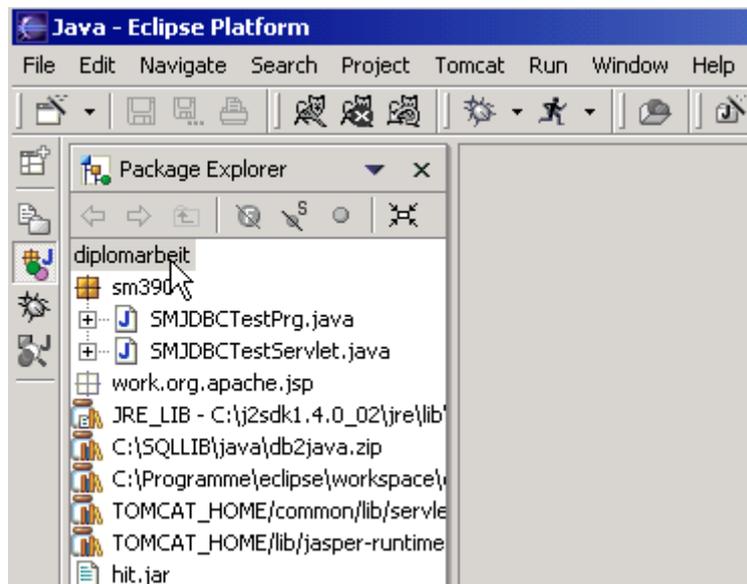
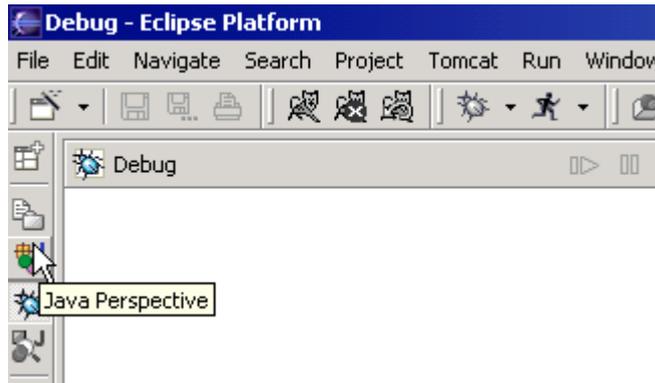
Klicken Sie im Eclipse Platform-Fenster links oben auf *File*, dann auf *exit*, um Eclipse zu beenden.

Nun können Sie alle noch offenen Fenster schließen und ihren PC herunterfahren.

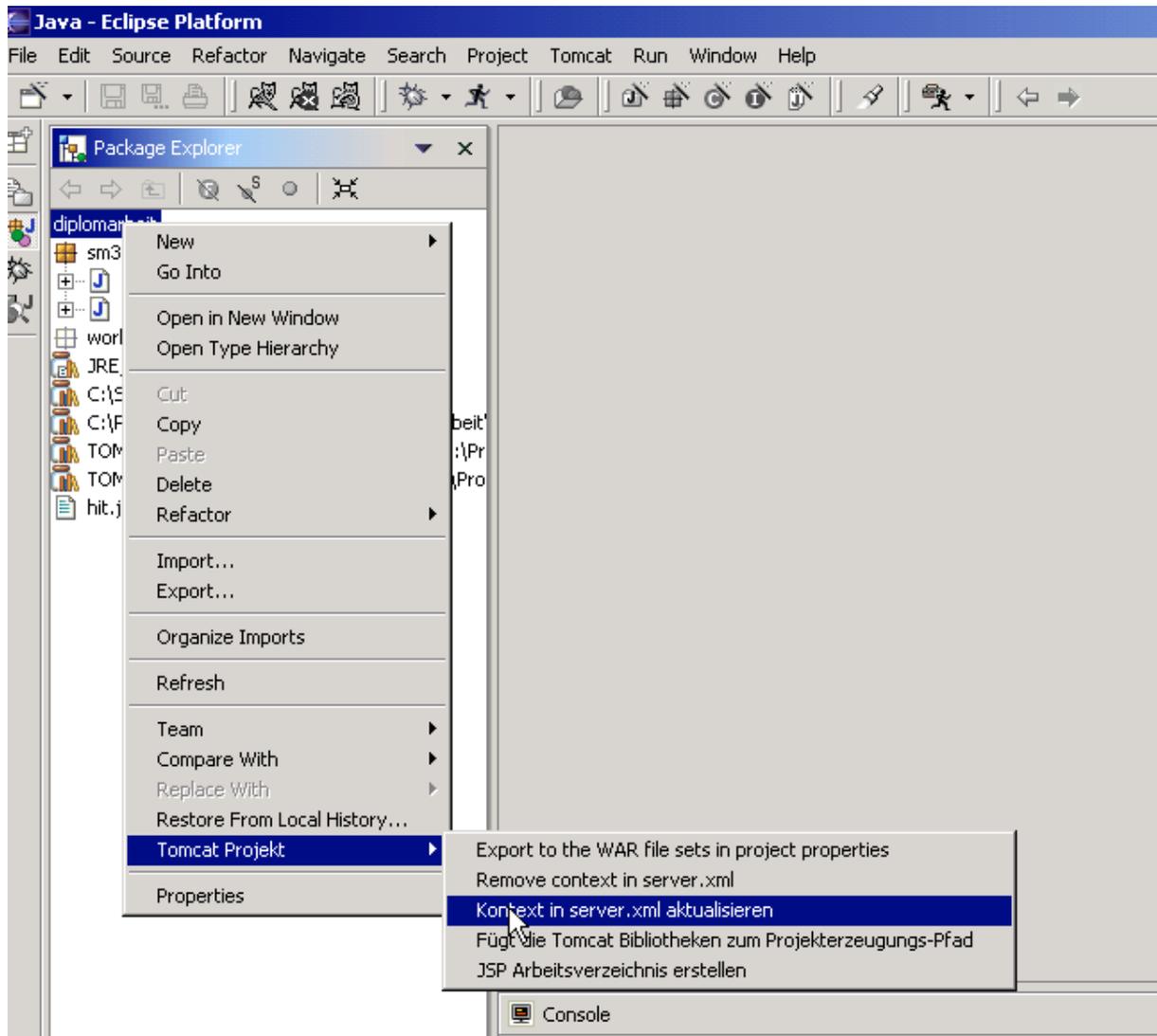
## 7. Erneute Inbetriebnahmen nach abgeschlossener Installation

Rechner hochfahren

Eclipse starten (Ordner *C:\Programme\eclipse* öffnen und 2k auf das Eclipse-Symbol)

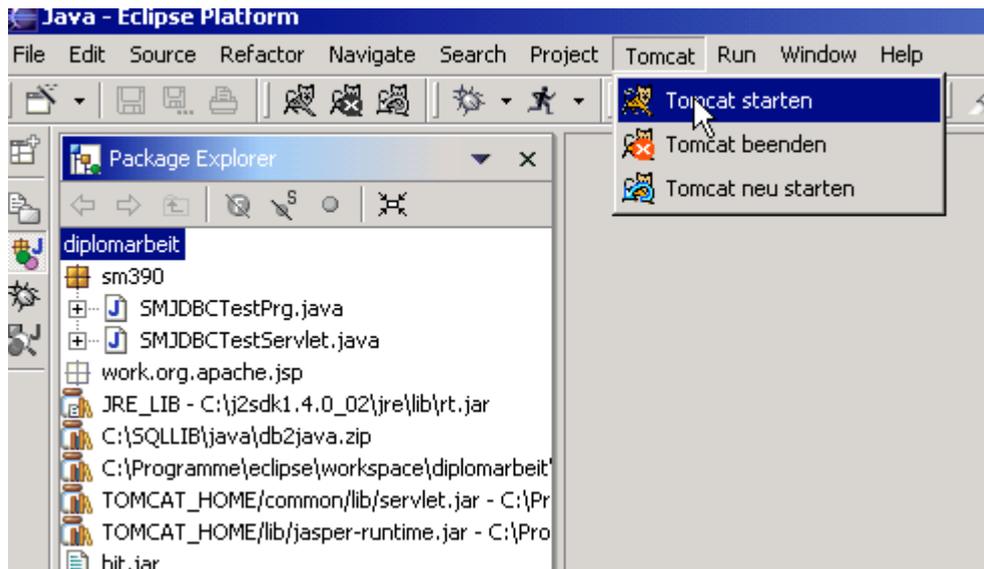
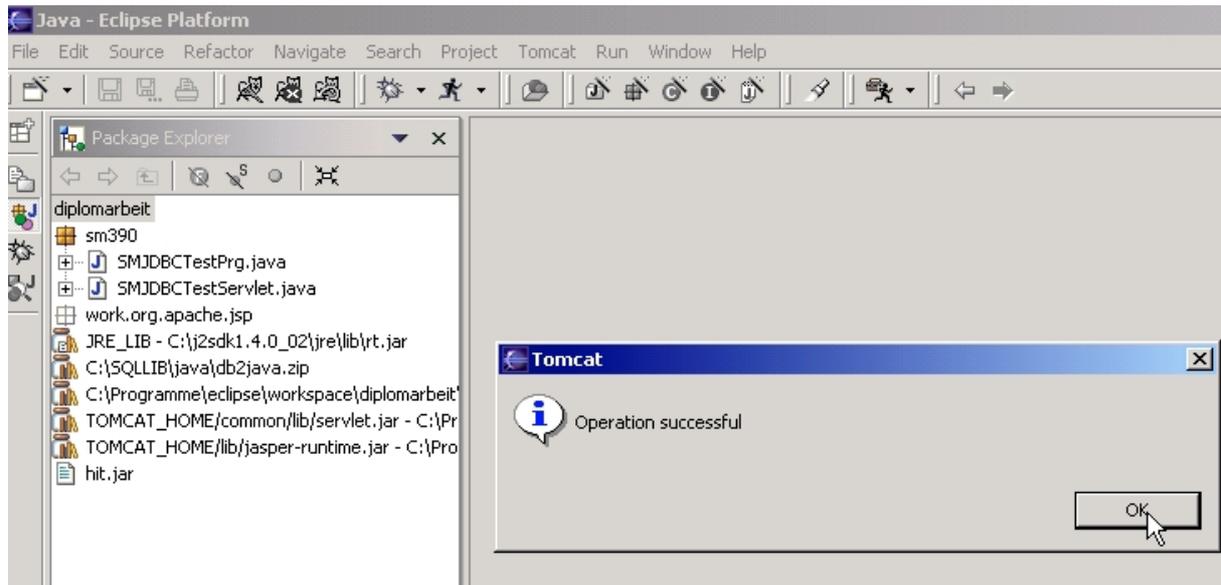


1kr auf *diplomarbeit*



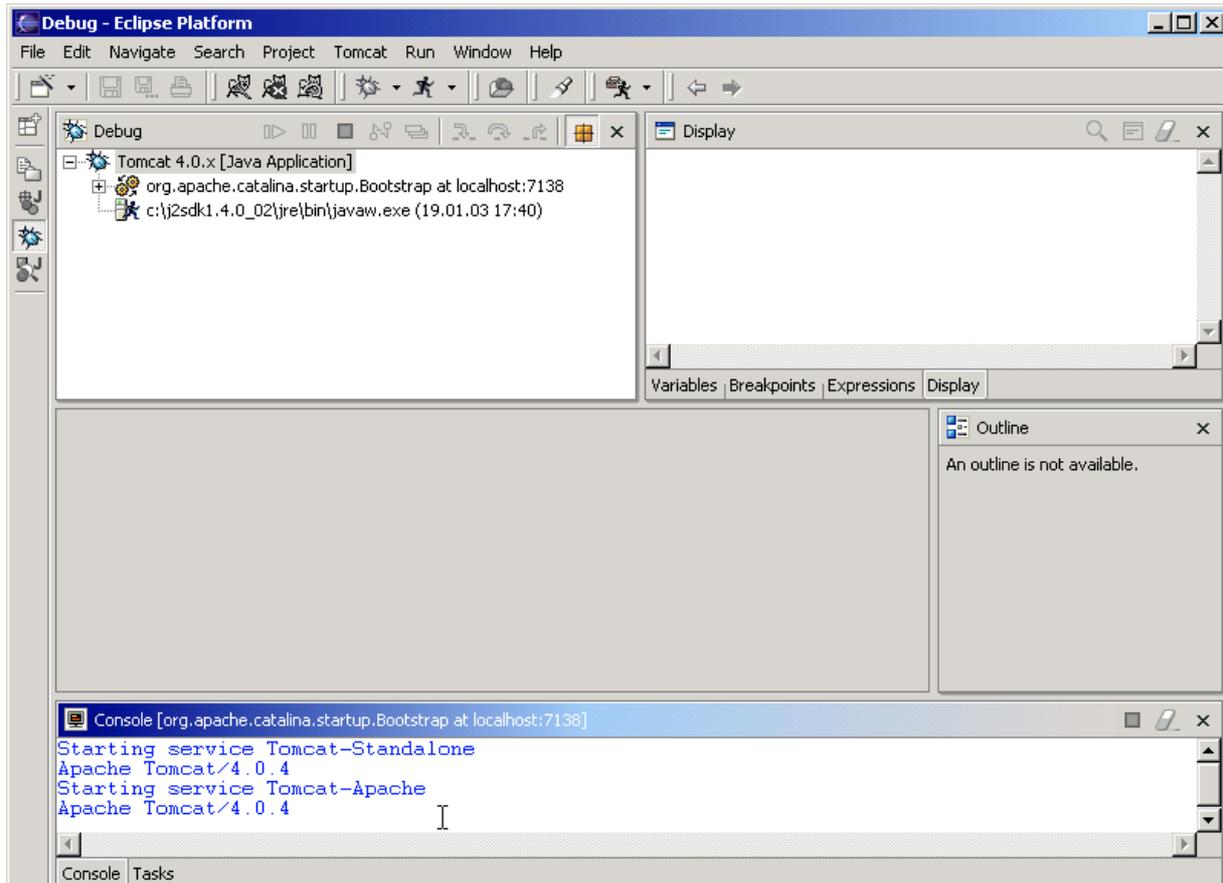
1kr auf *diplomarbeit*, *Tomcat Projekt* auswählen, *Kontext in Server .XML aktualisieren*, 1k

Es erscheint in einem sich öffnenden Fenster die Meldung "Operation successful".

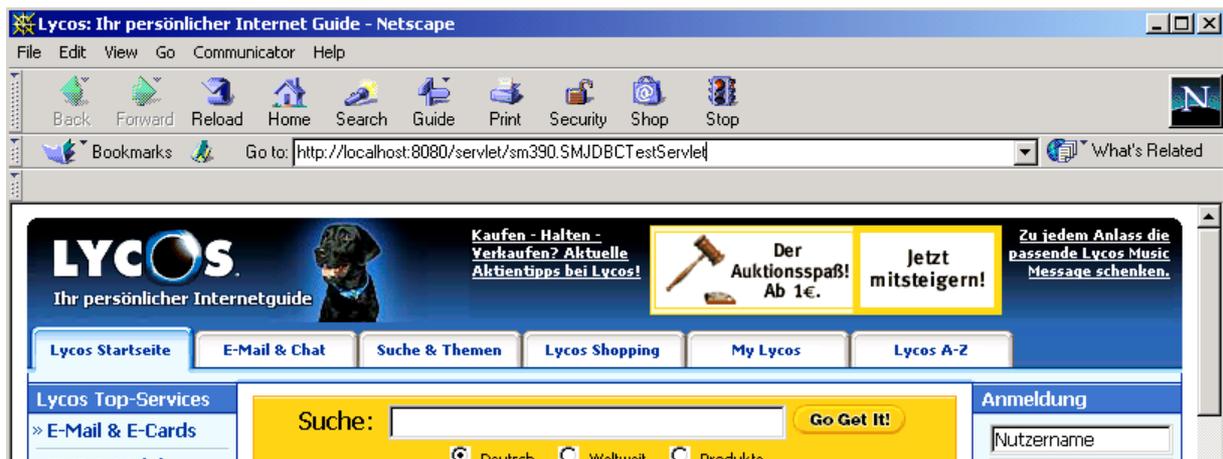


Möchte man Tomcat erneut starten:

1k auf *Tomcat*, einen weiteren auf *Tomcat starten*.

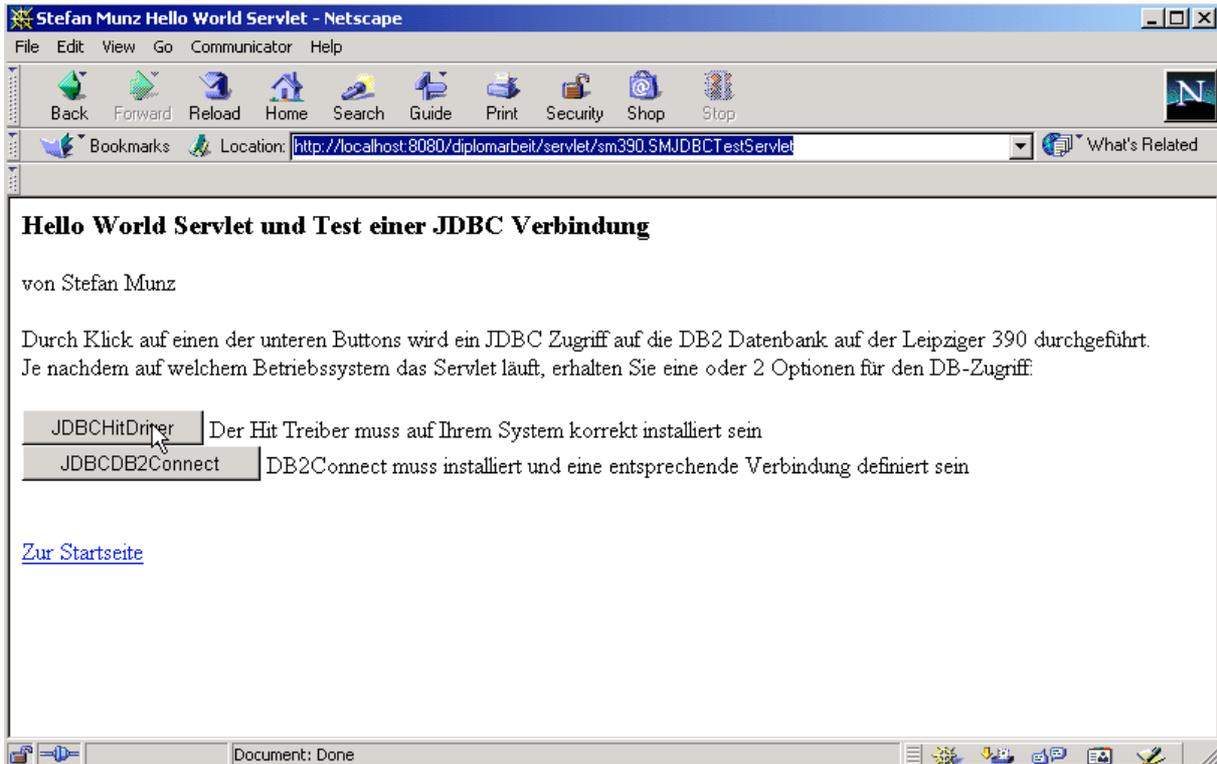


Warten, bis im unteren Fenster die hier gezeigten vier Zeilen erscheinen



Die URL `http://localhost:8080/servlet/sm390.SMJDBCTestServlet` in einen Browser eingeben.

Die Eingabetaste betätigen und kurz warten.



Im sich geöffneten Servlet-Fenster wieder auf den Button *JDBCHitDriver* klicken.

Die DB2-Tabelle auf dem OS/390-Server wird wieder ausgelesen:

