

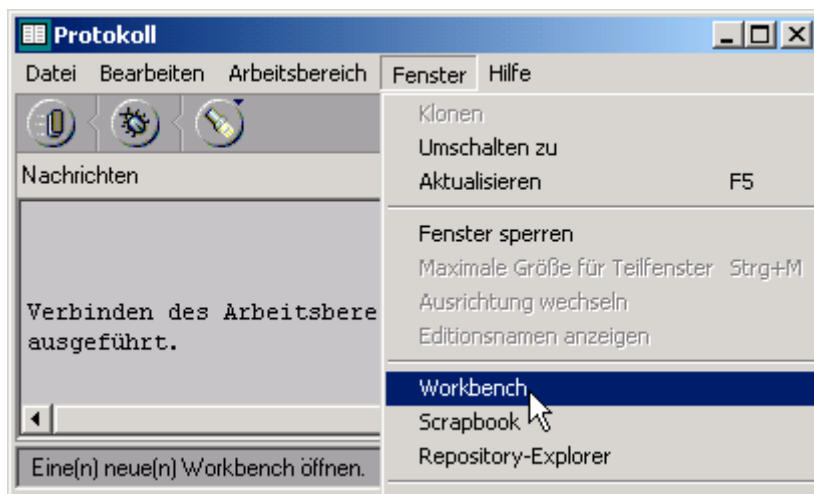
## Tutorial 9

### Zugriff auf eine DB2-Datenbank über DB2Connect oder JDBC mittels Java

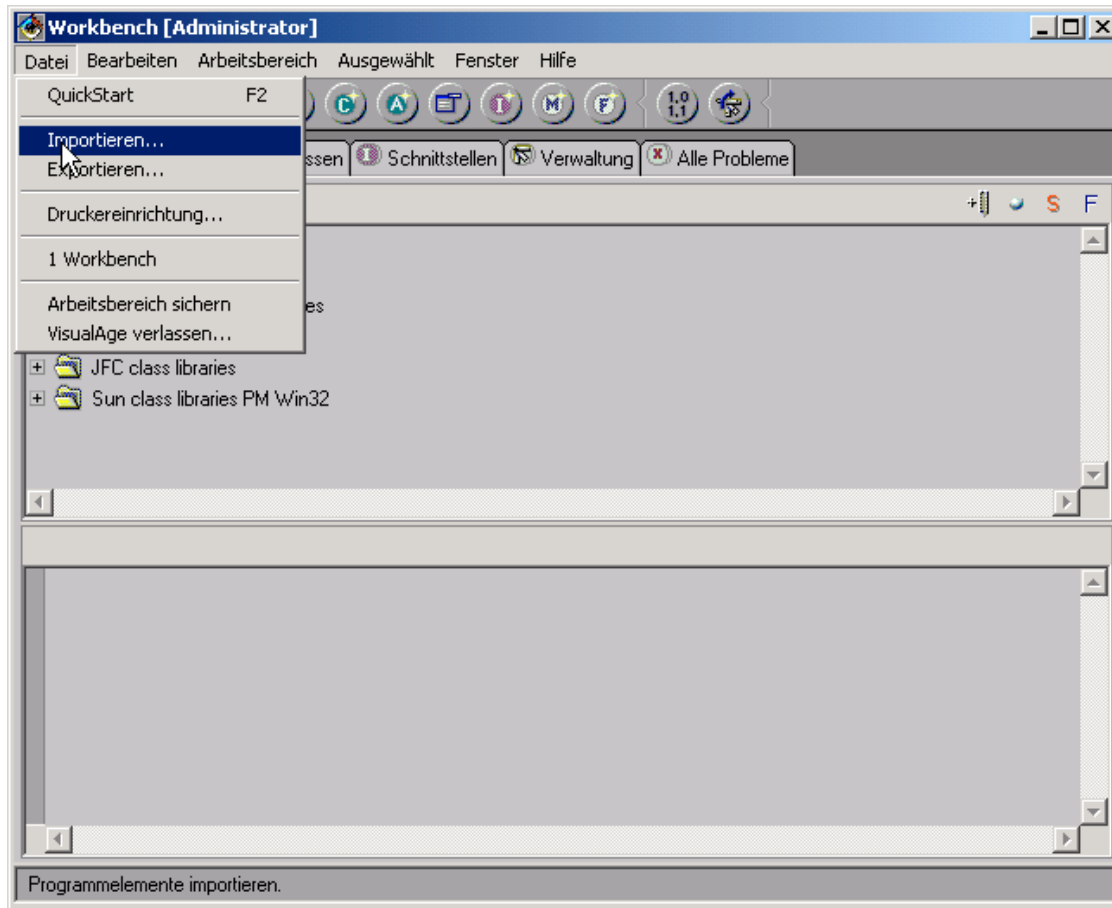
Ziel dieses Tutorials ist es, mit Hilfe eines vorgefertigten Java-Programmes von unserem Windows-Rechner einen Zugriff auf eine OS/390-Datenbank zu machen. Dazu ist die erfolgreiche Bearbeitung von Tutorial 8 notwendig.

*Aufgabe:* Arbeiten Sie das nachfolgende Tutorial durch.

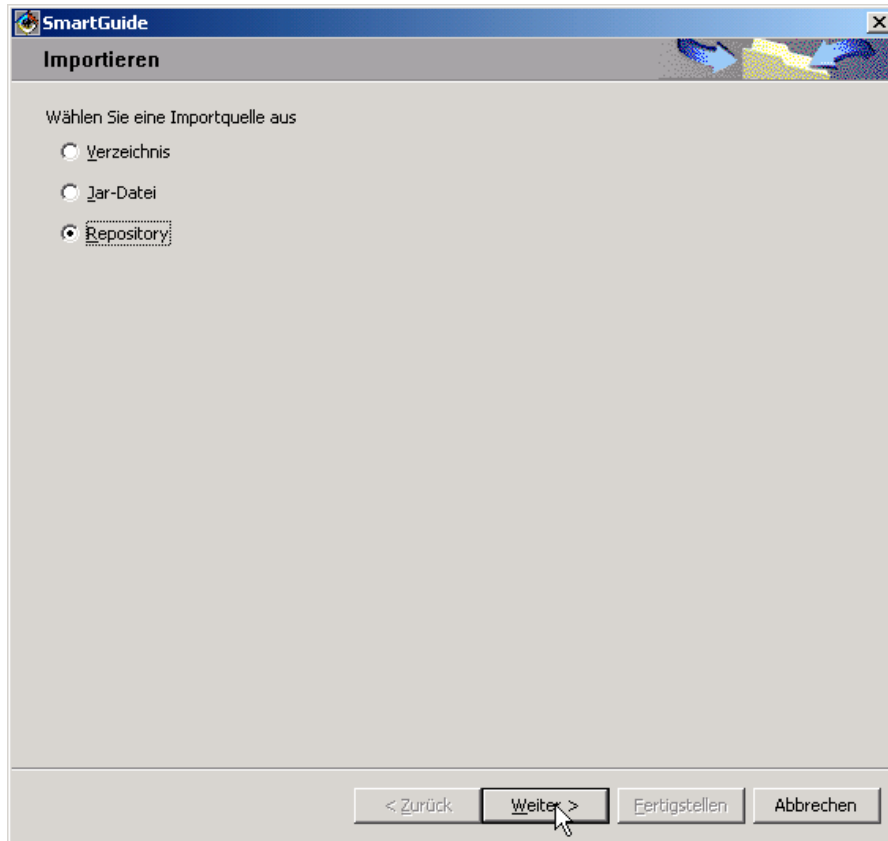
Der Zugriff soll einmal über DB2Connect und einmal über JDBC geschehen. Starten Sie ihren Windows 2000-Rechner mit der User-ID "db2admin" und rufen Sie anschließend Visual Age für Java auf.



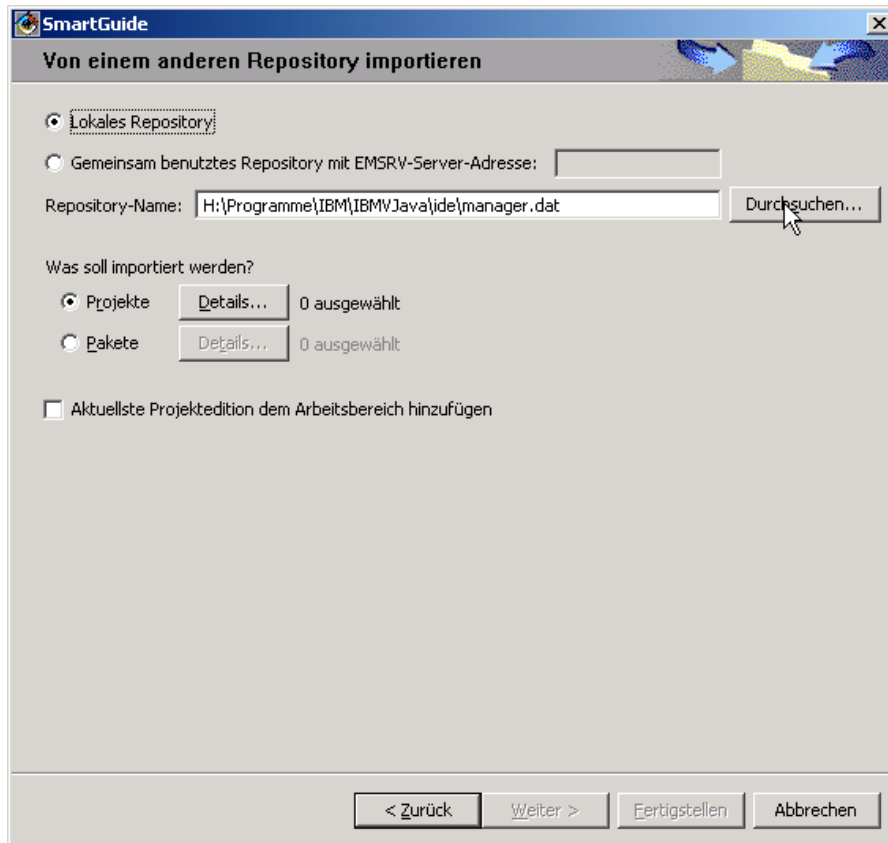
Falls das obige Fenster erscheint, ist in der Menüleiste auf "Fenster" zu klicken. Anschließend ist "Workbench" auszuwählen.



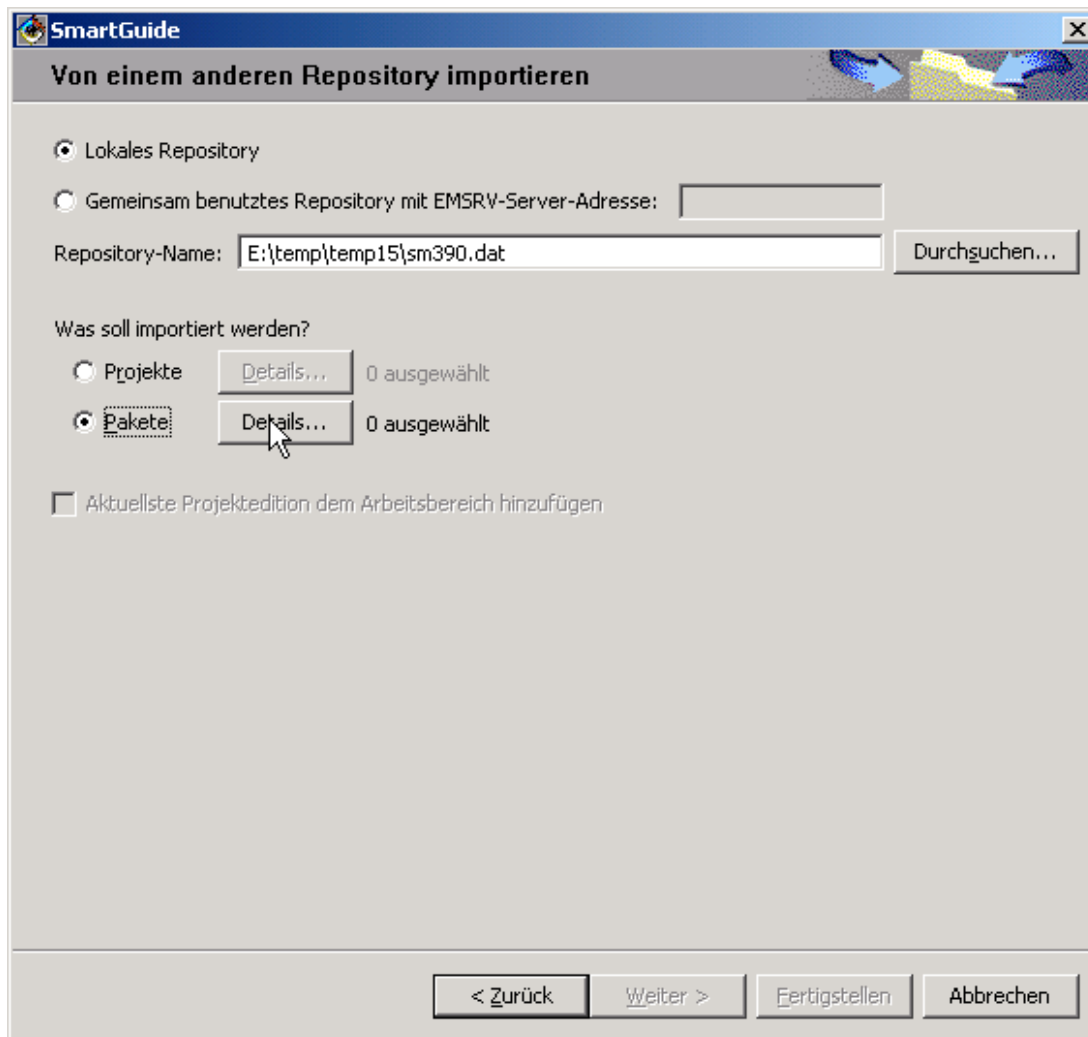
Die VisualAge Workbench erscheint. Jetzt wollen wir die .dat file importieren. Dazu ist in der Menüleiste auf "Datei" und anschließend auf "importieren" zu klicken.



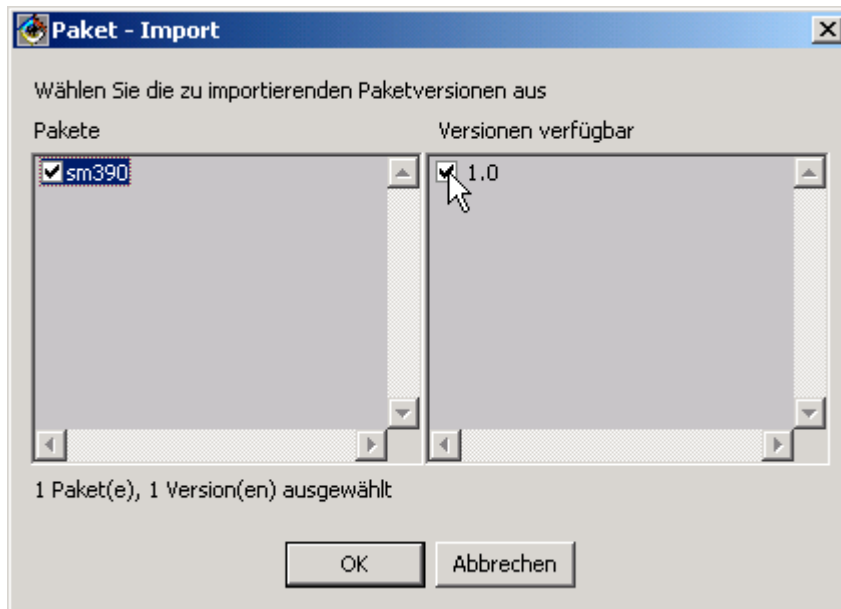
Wie oben dargestellt, ist erst auf "Repository", dann auf "Weiter" zu klicken.



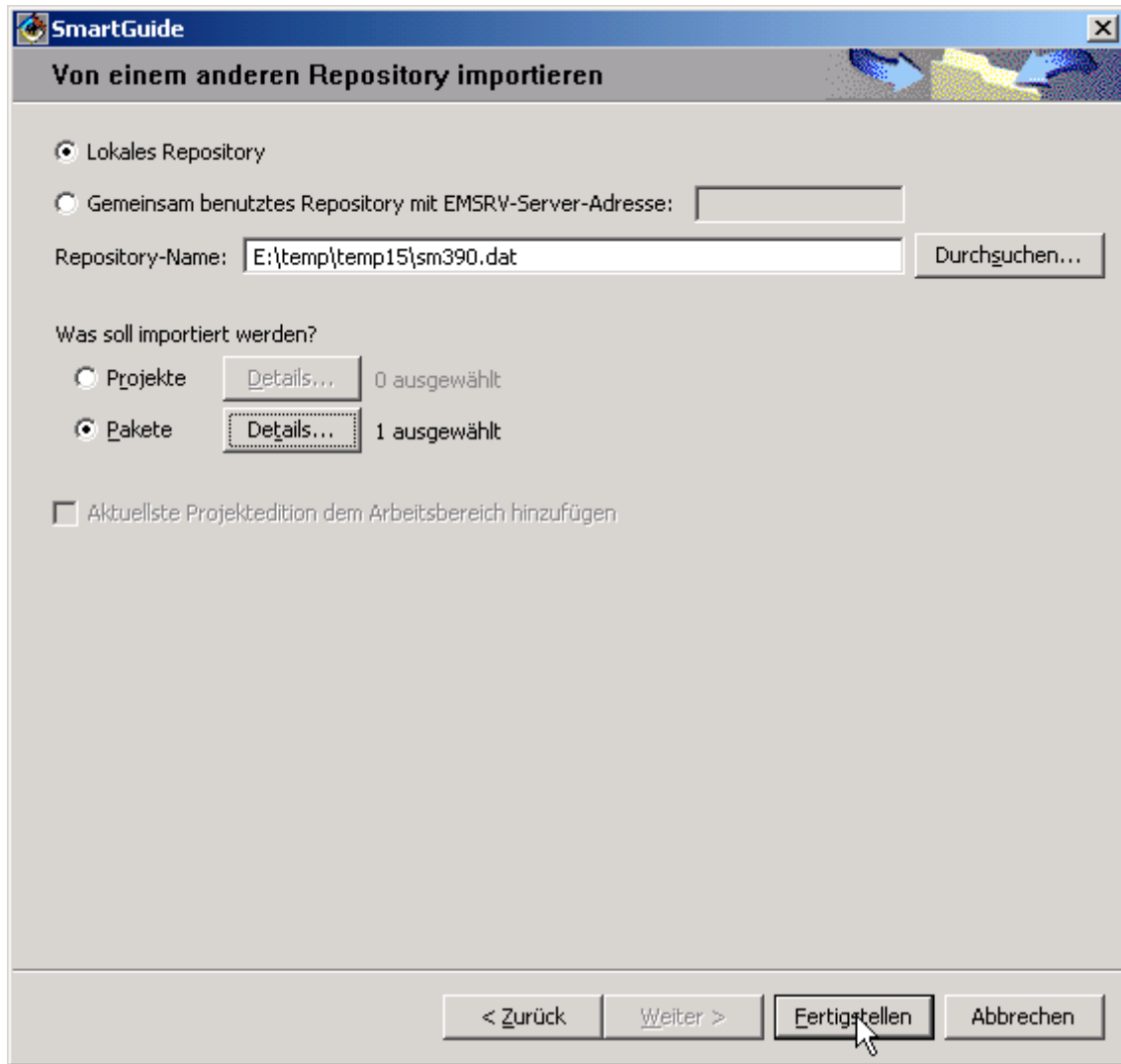
Wie im obigen Fenster dargestellt, ist zuerst "Lokales Repository" auszuwählen und anschließend auf "Durchsuchen" zu klicken. Wie im nachfolgenden Screen dargestellt, ist der Pfad der Datei "sm390.dat" aufzusuchen.



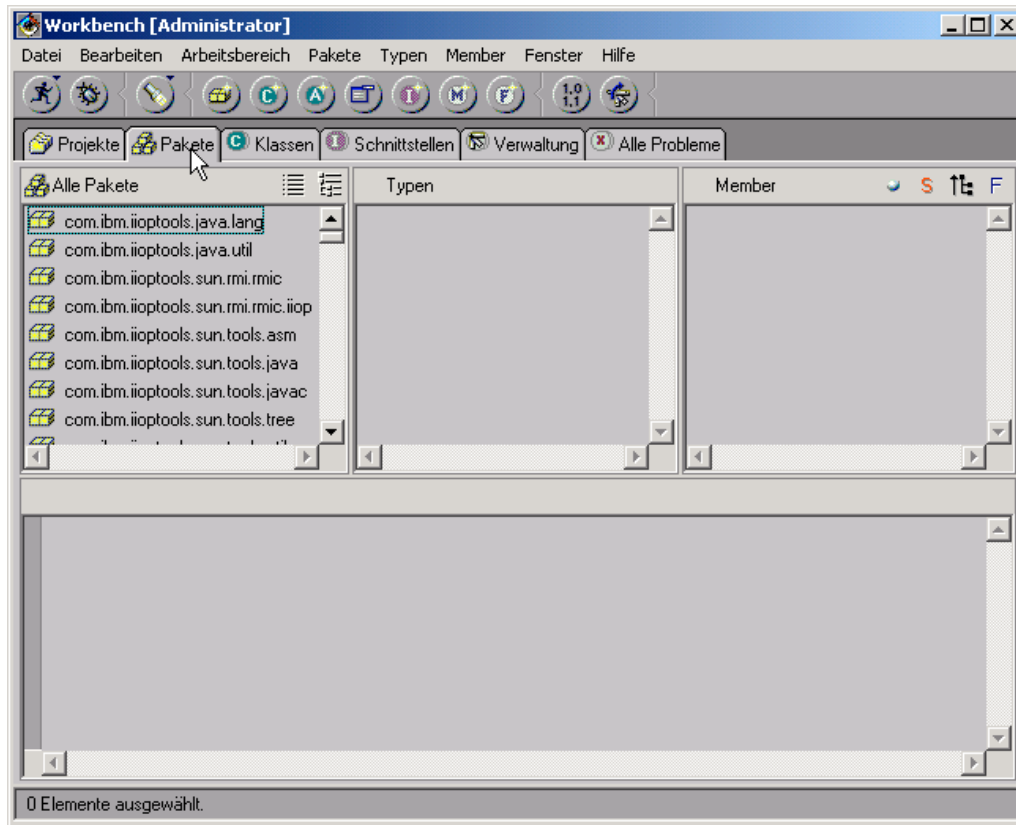
Nun muß noch eingestellt werden, was importiert werden soll. Dazu ist auf "Pakete" und anschließend auf "Details" zu klicken.



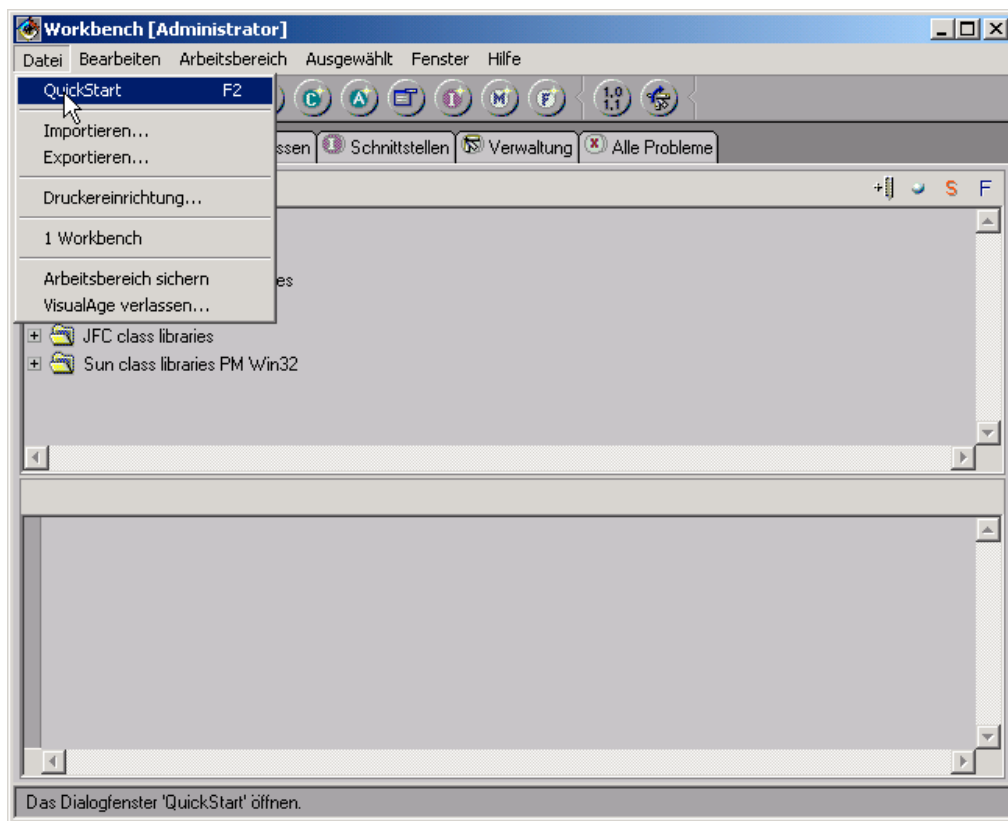
Wie obige zwei Abbildungen zeigen, ist auf das kleine weiße Quadrat neben "sm390" zuklicken, so dass in diesem ein Häkchen erscheint. Wenn notwendig, ist auch für das Häkchen vor der Versions-Nummer "1.0" zu sorgen.



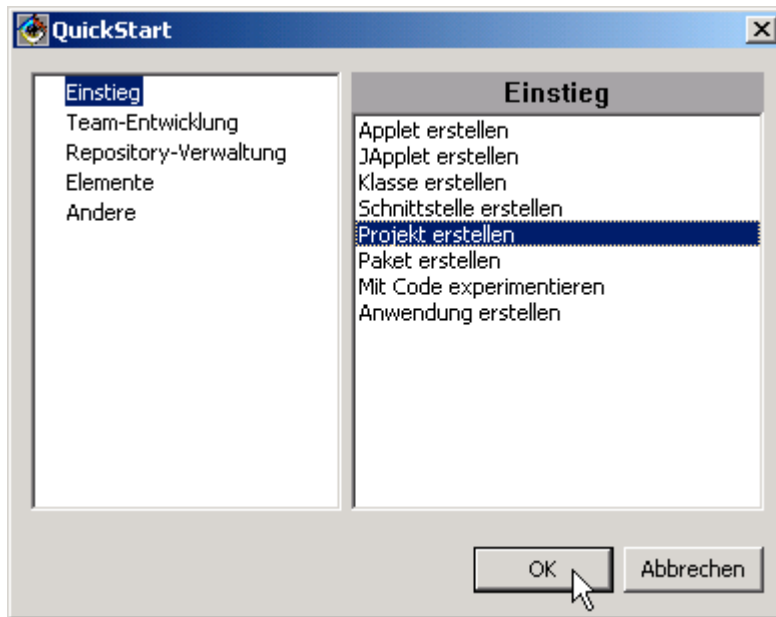
Durch Klick auf "Fertigstellen" wird der Import von "sm390.dat" aus dem lokalen Repository abgeschlossen.



Das Workbench-Fenster erscheint wieder. Wenn wir die Registerkarte "Pakete" betätigen, erscheinen dort alle für die Workbench verfügbaren Pakete. Unser Paket "munz390" ist aber nicht darunter. Damit wir damit arbeiten können, müssen wir zunächst ein Projekt erstellen.

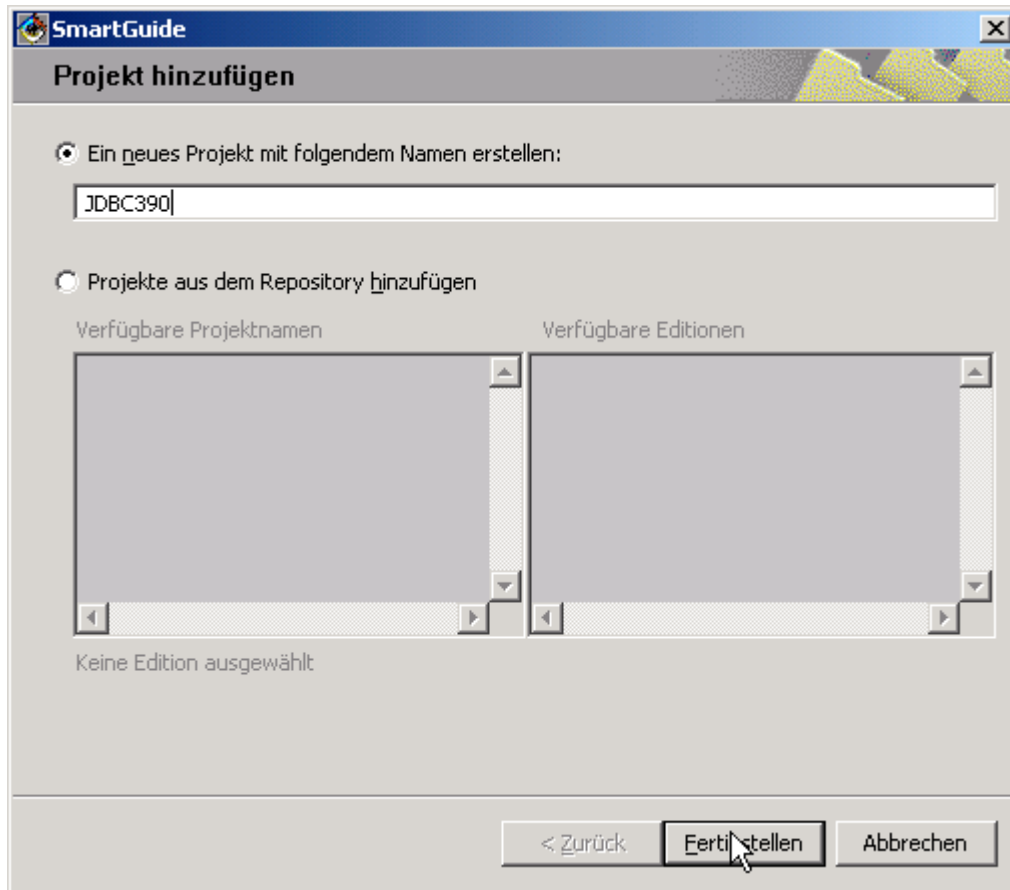


Wie oben dargestellt, ist in der Menüleiste "Datei" und anschließend "QuickStart" auszuwählen.

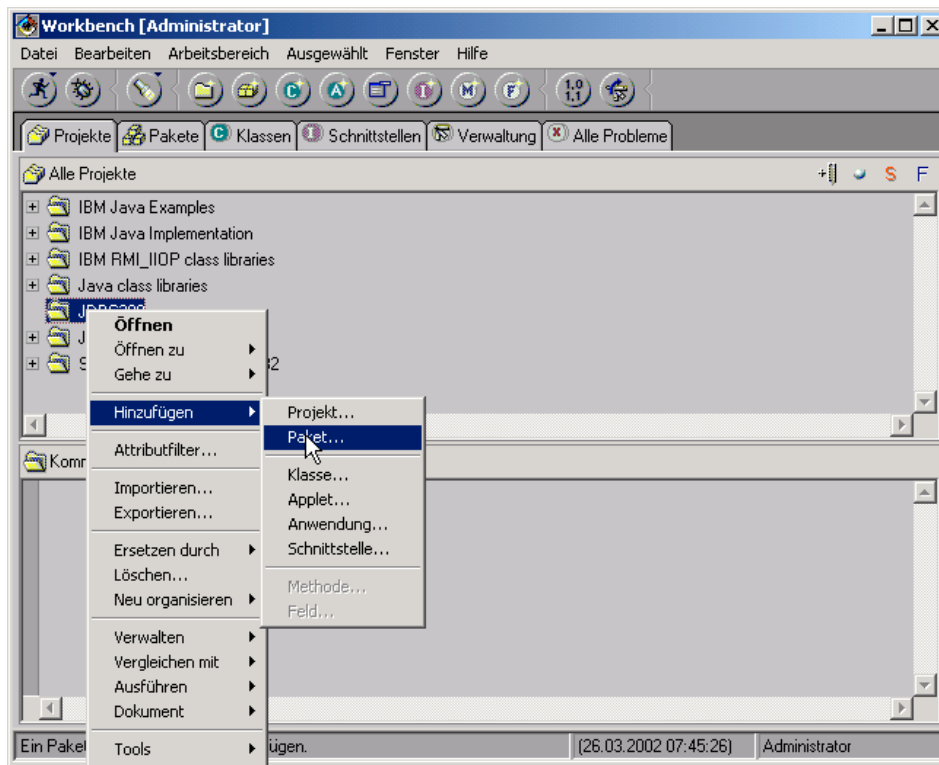


Es erscheint das obige QuickStart-Fenster. Hier sind "Einstieg" sowie "Projekt erstellen" auszuwählen und diese Aktion mit Klick auf "OK" abzuschließen

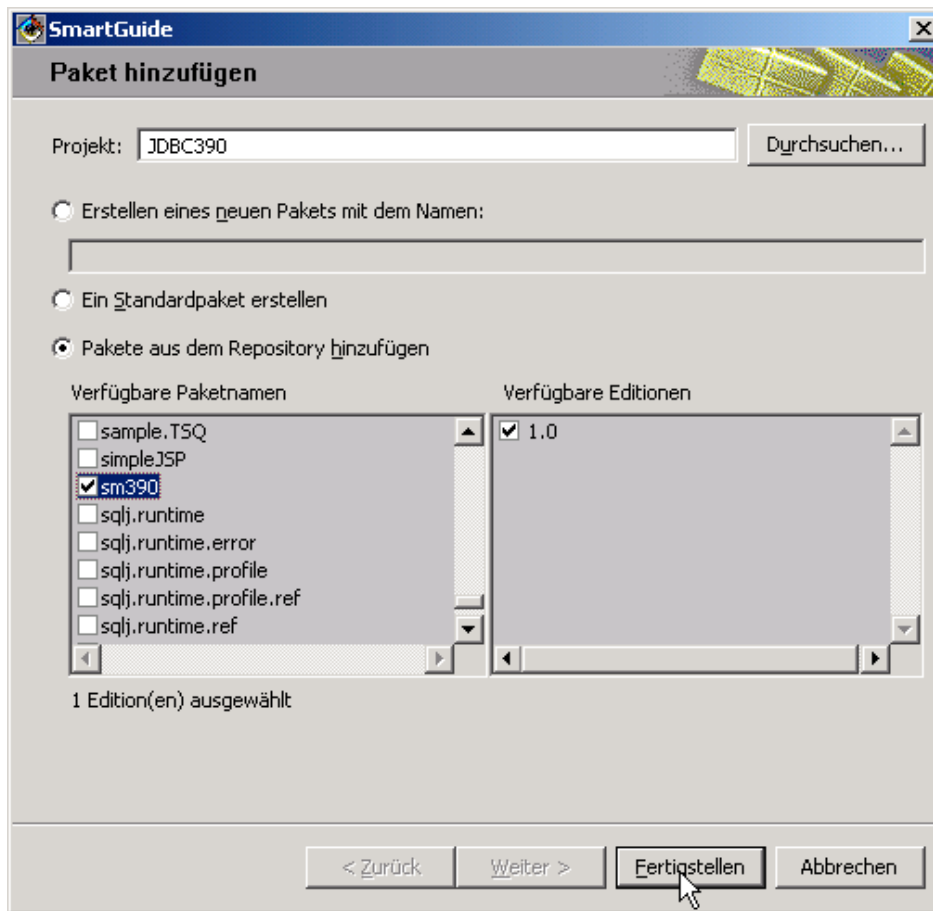




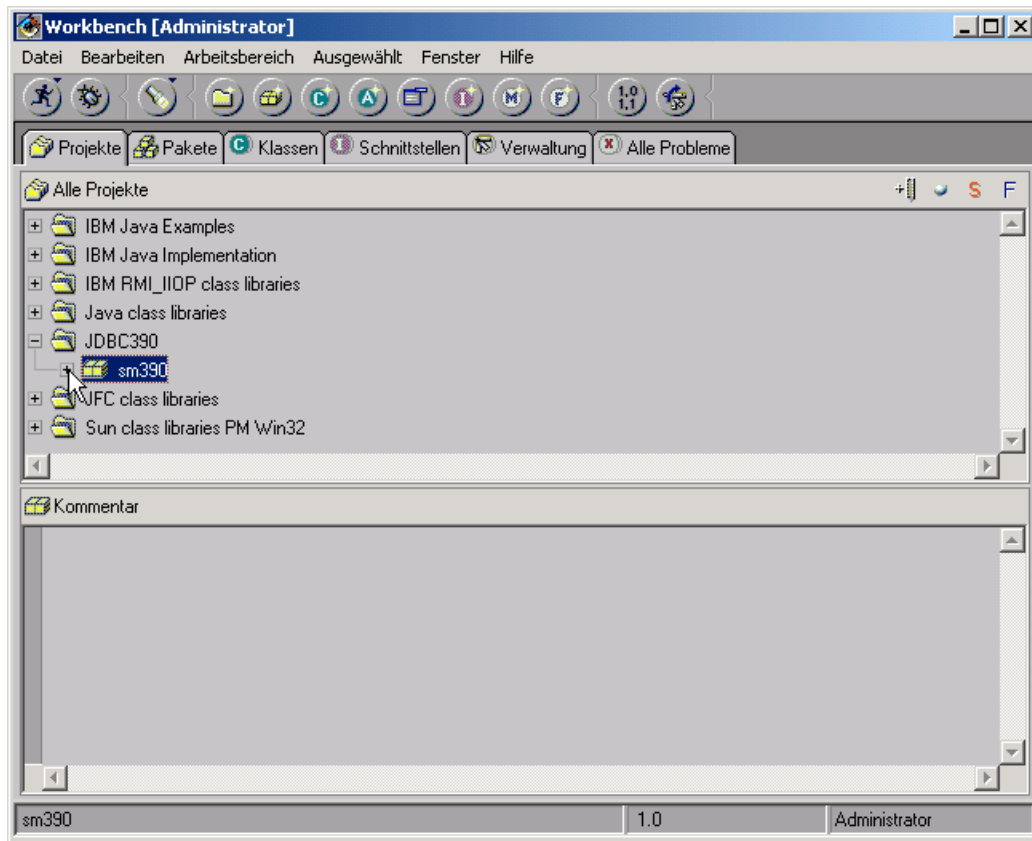
Als Name für das neu zu erstellende Projekt kann man "JDBC390" oder einen anderen Namen wählen. Diese Aktion ist durch Klick auf "Fertigstellen" abzuschließen.



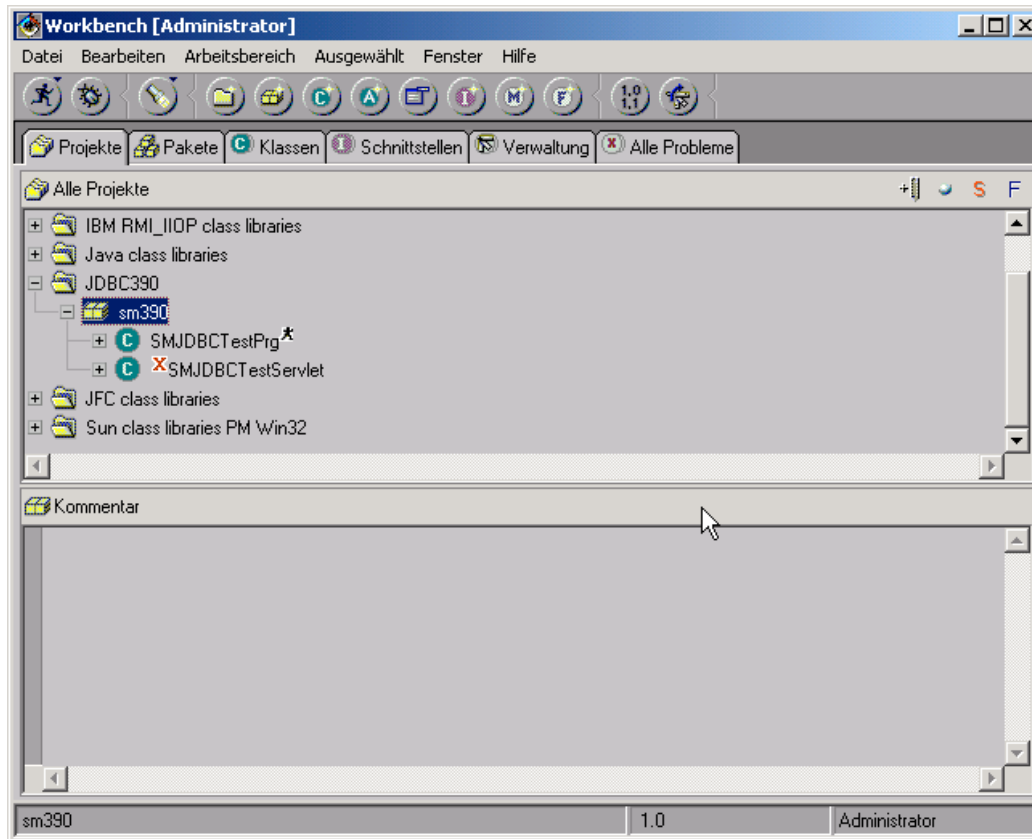
Im Workbench-Fenster erscheint in der Registerkarte "Projekte" das neue Projekt "JDBC". Wie oben dargestellt ist auf das Projekt mit der rechten Maustaste zu klicken. Anschließend sind "Hinzufügen" sowie "Paket ..." auszuwählen.






Wählen Sie "Pakete aus dem Repository hinzufügen" aus und suchen Sie, wie oben dargestellt, unter den verfügbaren Paketen "sm390" heraus, indem Sie vor "sm390" das Häkchen setzen. Klicken Sie abschließend auf "Fertigstellen".



Das Workbench-Fenster erscheint wieder. Unter dem Projekt "JDBC390" wird jetzt das Paket "sm390" sichtbar. Klicken Sie auf das Plus-Zeichen vor "sm390".



Unter "sm390" werden jetzt 2 Klassen sichtbar: "SMJDBCTestPrg" und "SMJDBCTestServlet". Diese Klassen waren in dem von uns importierten Paker "sm390.dat" enthalten.

Das Symbol  bedeutet, dass es sich um eine Java-Klasse handelt. Das -Symbol besagt, dass die Klasse ausführbar ist.  besagt, dass der Klasse noch irgendetwas fehlt.

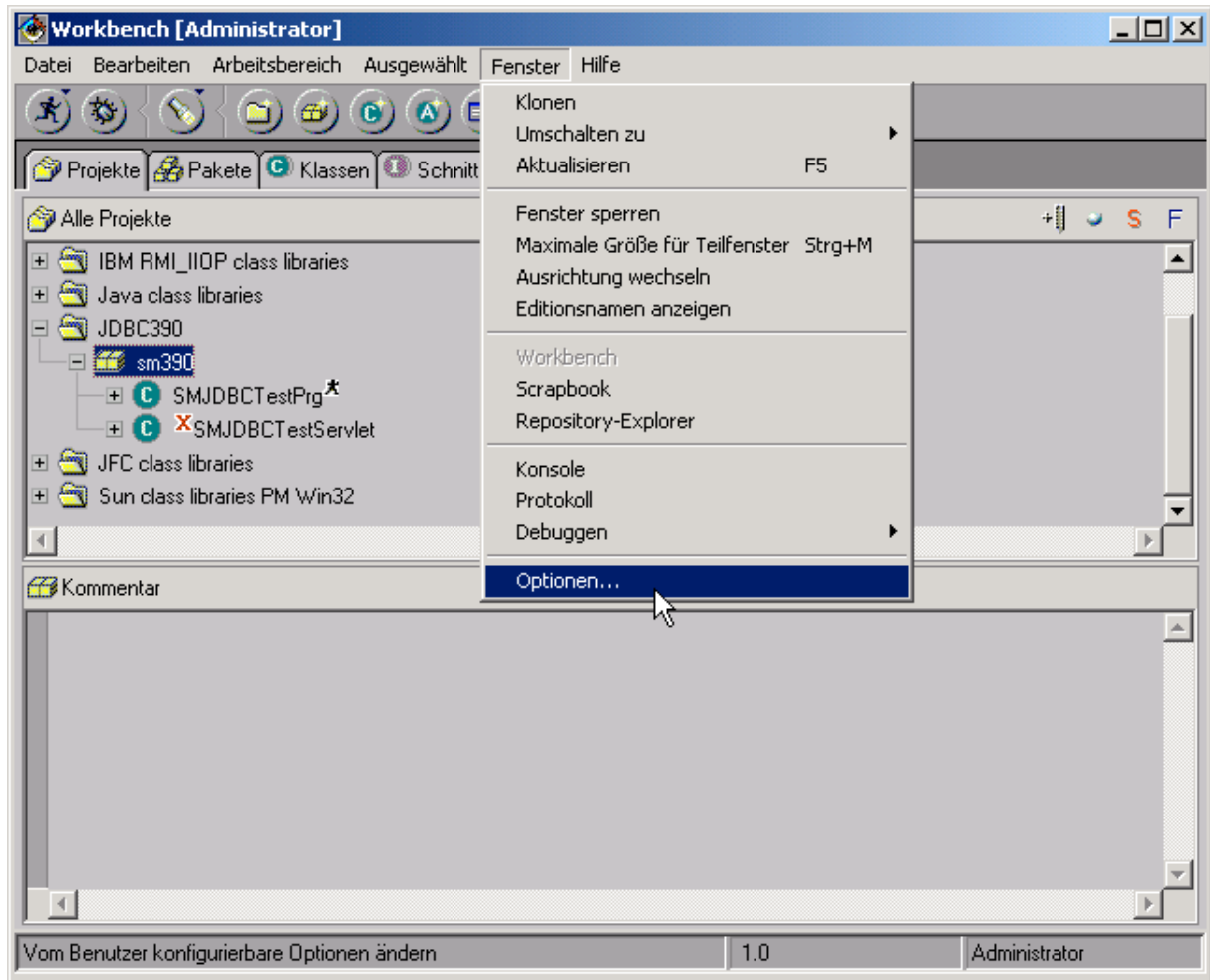
Wir haben damit die Voraussetzung geschaffen, mit Hilfe des Java-Programmes "SMJDBCTestPrg", welches auf unserem Windows 2000-Rechner ausgeführt wird, auf die OS/390 DB2-Datenbank mit Hilfe des DB2Connect Connectors zuzugreifen. Ehe wir dies austesten, wollen wir auch eine Verbindungsmöglichkeit mit einem JDBC Connector schaffen.

## Arbeiten mit JDBC

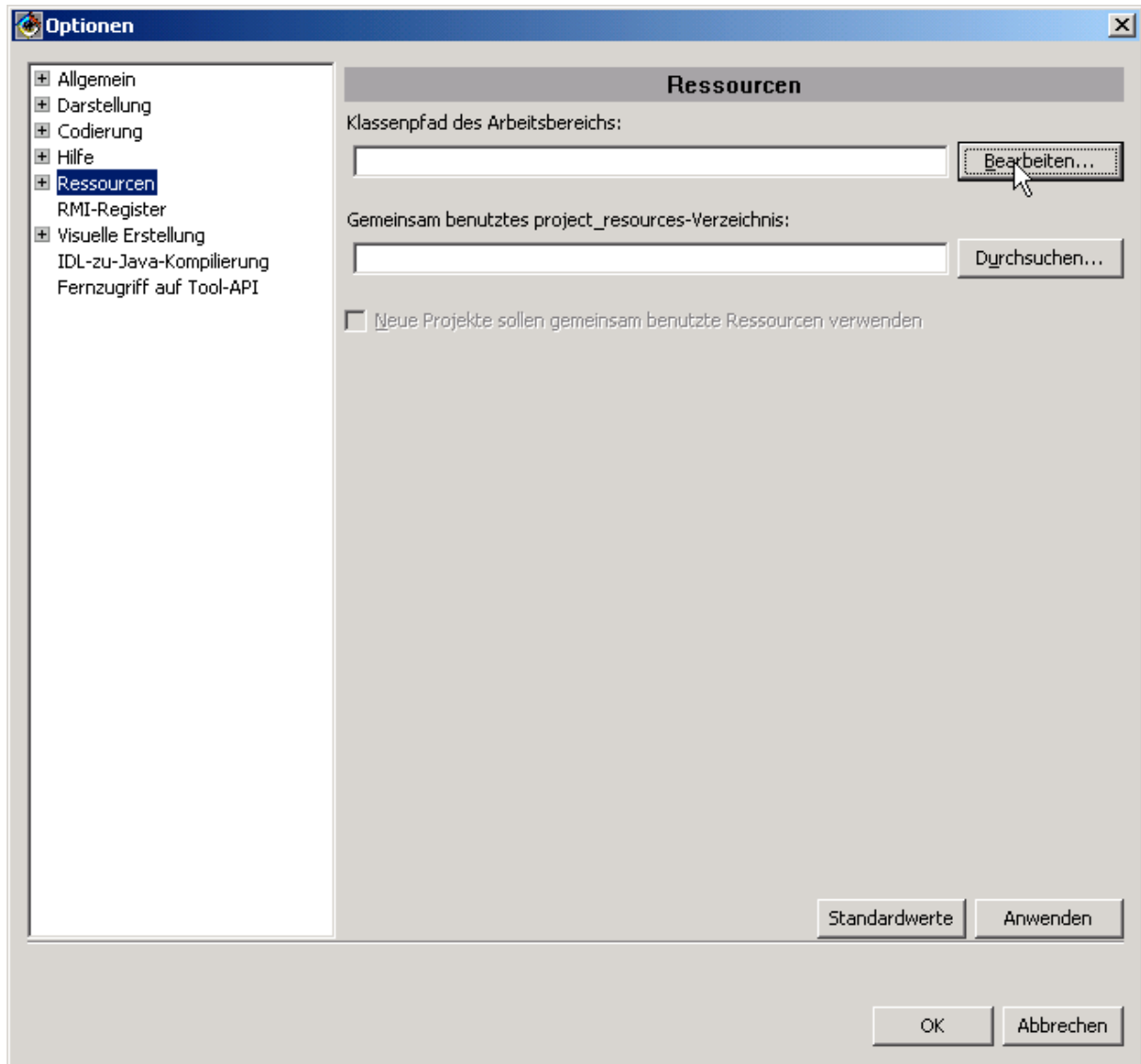
Um diesen Zugriff mit JDBC durchführen zu können, brauchen wir einen JDBC Connector. Wir benutzen hierzu den Connector der Firma Hit. Er steht uns als File "hit.zip" zur Verfügung.

Wir erzeugen ein neues Verzeichnis "H:\driver\hit" auf unserer Festplatte. Wenn Sie von CDs alle Tools für die Tutorien 8 und 9 installiert haben, dann finden Sie auf der Installations-CD 2/3 ein File "hit.zip". Wenn Sie die Tools per FTP heruntergeladen haben und anschließend entpackt haben, dann finden Sie dieses File "hit.zip" im Verzeichnis "C:\TUTOR08" auf ihrer Festplatte. Dieses enthält in gepackter Form 10 Verzeichnisse mit Class-Dateien. Wir entpacken "hit.zip" (mittels WinZip oder WinAce) bei Übernahme der Verzeichnisstruktur in das gerade angelegte Verzeichnis "H:\driver\hit". Als Ergebnis müssten in "H:\driver\hit" zehn Unterverzeichnisse entstehen, welche alle mit Class-Dateien gefüllt sind.

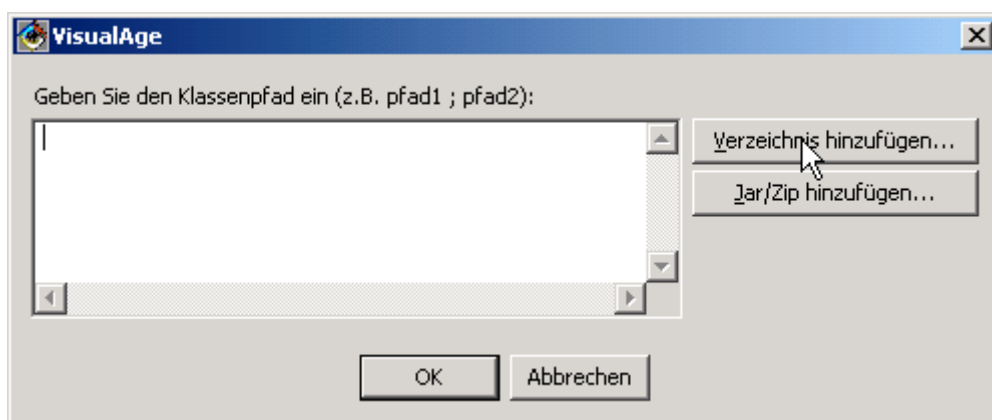
Als nächstes muss der Hit Connector in den Classpath aufgenommen werden.



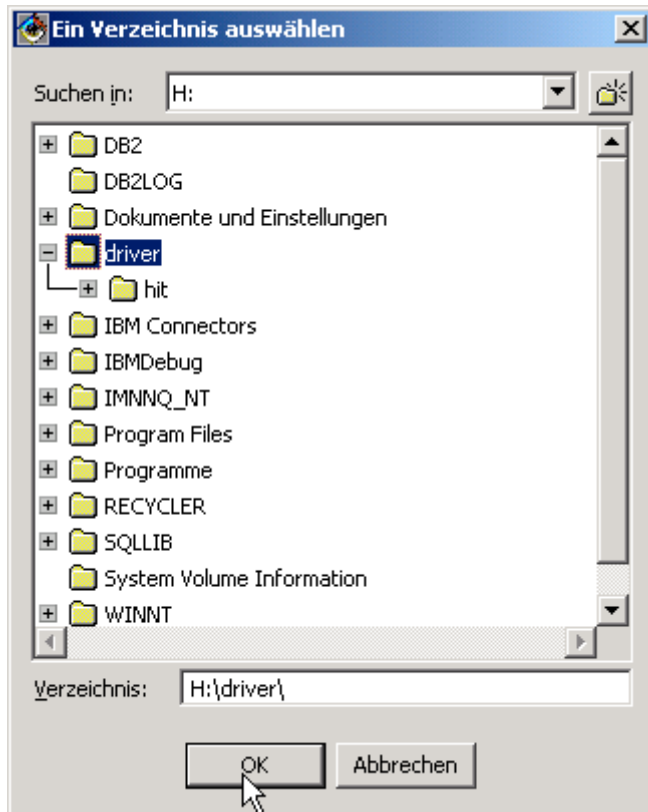
Dazu ist wie oben dargestellt "Fenster" und danach "Optionen..." auszuwählen.



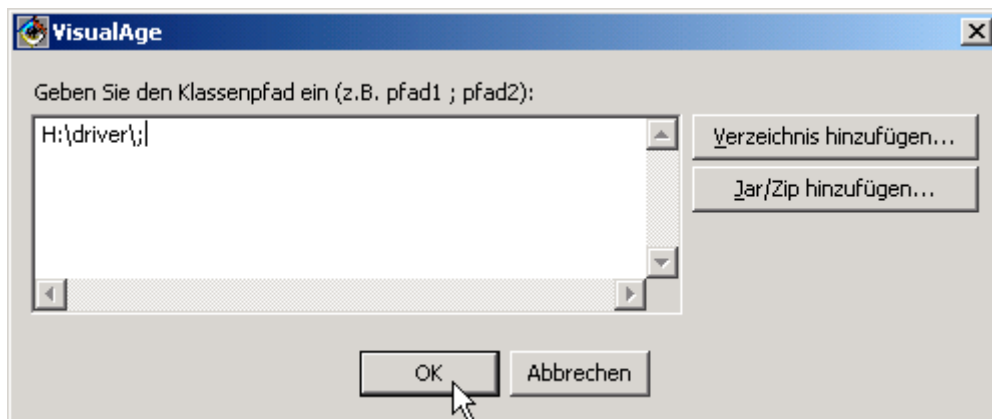
Das Fenster "Ressourcen" erscheint. Jetzt ist der Klassenpfad des Arbeitsbereiches zu bearbeiten. Klicken Sie dazu auf den Button "Bearbeiten..."



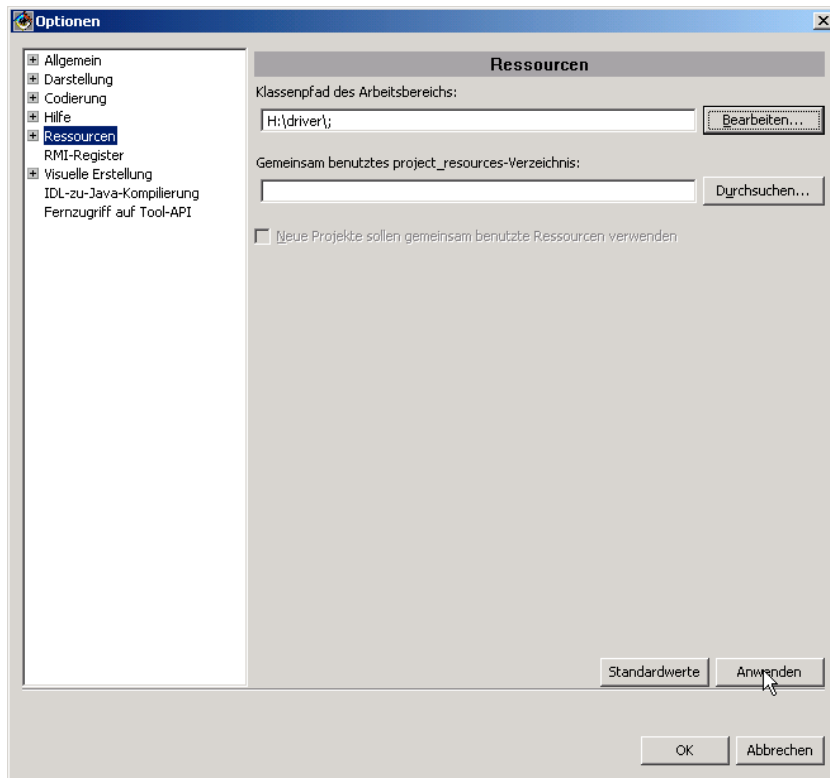
Im sich öffnenden Fenster Klick auf den Button "Verzeichnis hinzufügen..."



Jetzt ist das Verzeichnis des JDBC Connectors anzugeben oder auszuwählen, in diesem konkreten Beispiel also "H:\driver\". Diese Aktion ist mit Klick auf "OK" zu beenden.

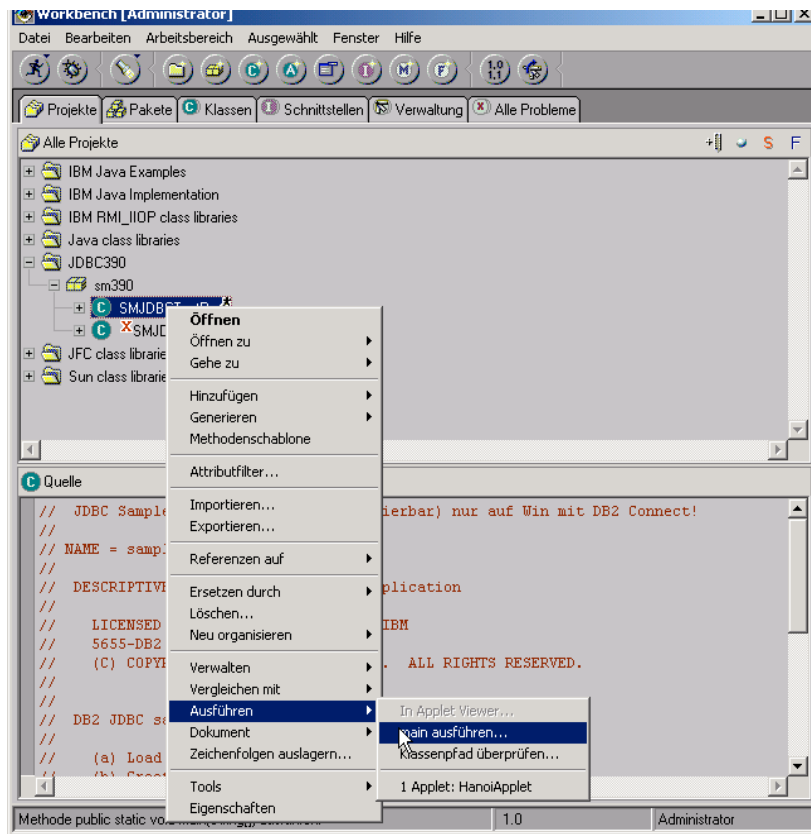


"H:\ driver" steht jetzt im Klassenpfad. Klicken Sie nun auf "OK".



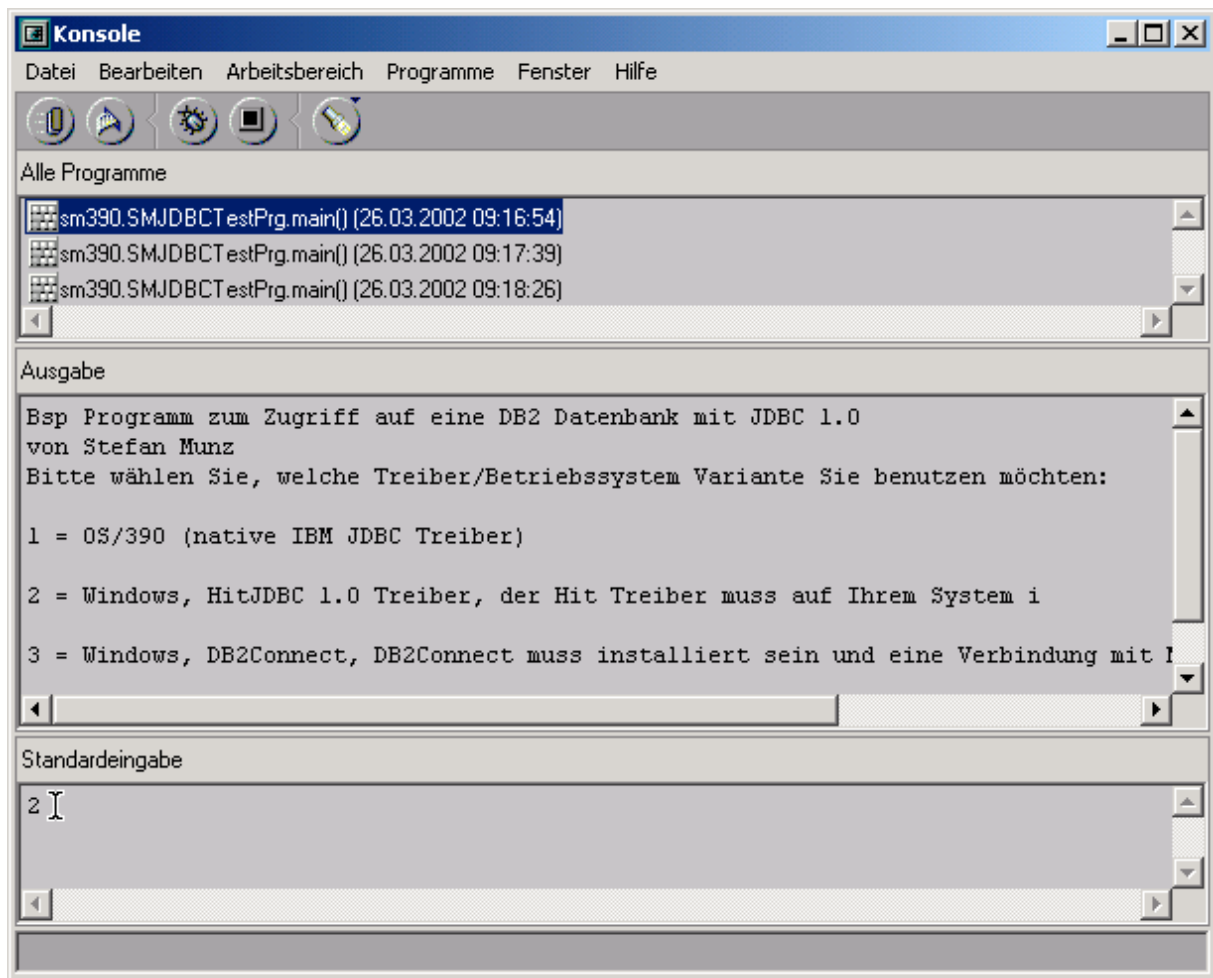
Im obigen Fenster ist zuerst auf "Anwenden" und danach auf "OK" zu klicken.

Das Workbench-Fenster erscheint wieder. Damit sind wir nun bereit, unseren ersten Test mit dem Hit (JDBC) Connector zu durchzuführen.





Wir wollen das Java-Programm "SMJDBCTESTestPrg" ausführen. Wir klicken dazu mit der rechten Maustaste auf "SMJDBCTESTestPrg". Anschließend ist "Ausführen" sowie "main ausführen...", wie in obiger Abbildung dargestellt, auszuführen.

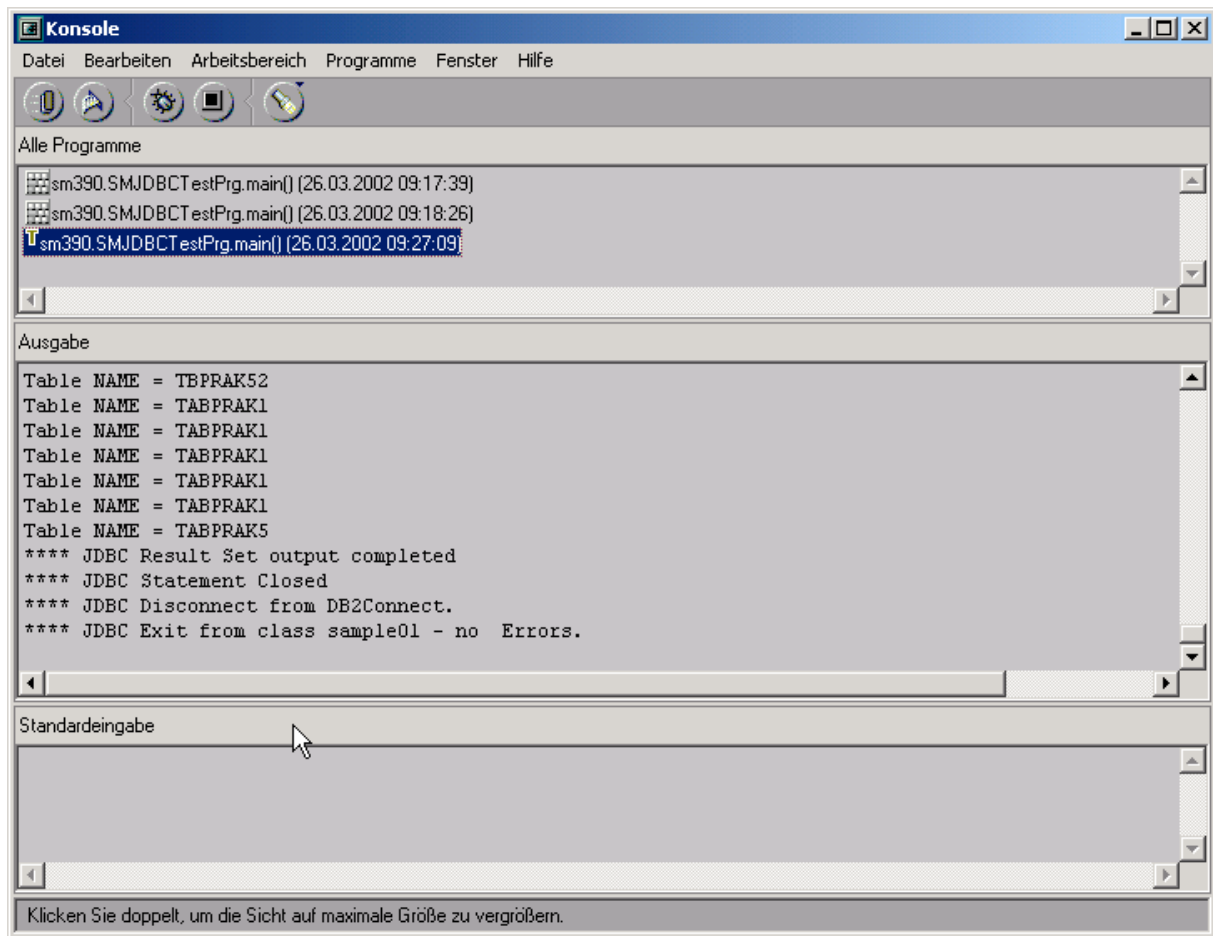


Das Konsole-Fenster erscheint. In dem Ausgabe-Unterfenster erscheint, was "SMJDBCTESTestPrg.main" auf den Bildschirm schreibt. In diesem Fall schreibt es die Anweisung, eine der drei Zahlen 1, 2 oder 3 einzugeben. Jenachdem, welche Zahl eingegeben wird, erfolgt ein DB2-Zugriff mit einem unterschiedlichen Treiber/Connector.

Das entsprechende Java Code-Fragment sieht wie folgt aus:

```
if (sDriver.equals("2"))
{
    Class.forName("hit.db2.Db2Driver");
    // Create the connection
    con =
    DriverManager.getConnection("jdbc:db2://jedi.informatik.uni-
    leipzig.de:446;rdbname=DBA1", user, password);
}
else
if (sDriver.equals("3"))
{
    Class.forName("COM.ibm.db2.jdbc.app.DB2Driver");
    // Create the connection
    con = DriverManager.getConnection("jdbc:db2:DBA1", user, password);
}
```

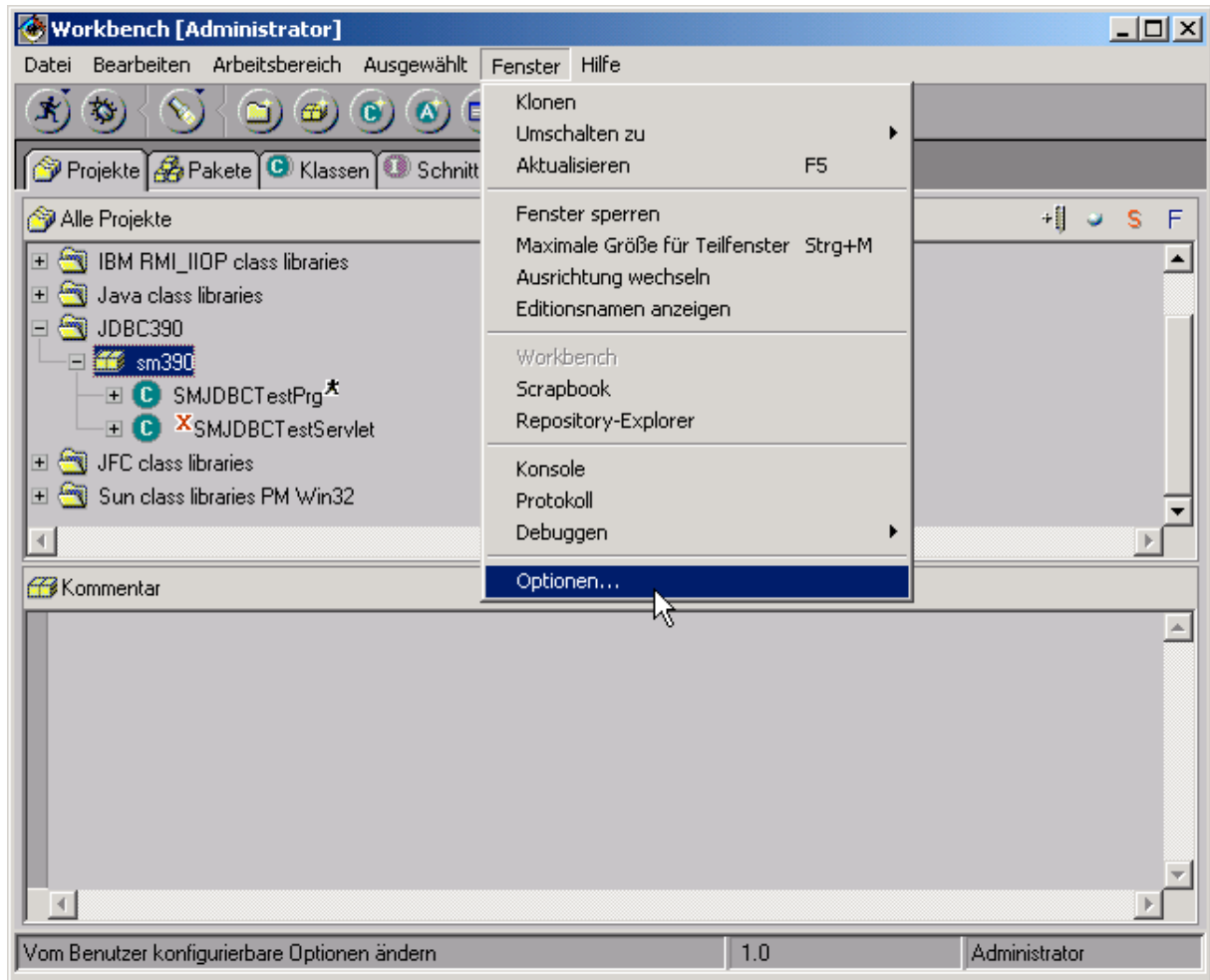
Wir wollen zunächst den JDBC Connector testen, und geben deshalb in dem Eingabe-Fenster eine "2" ein.



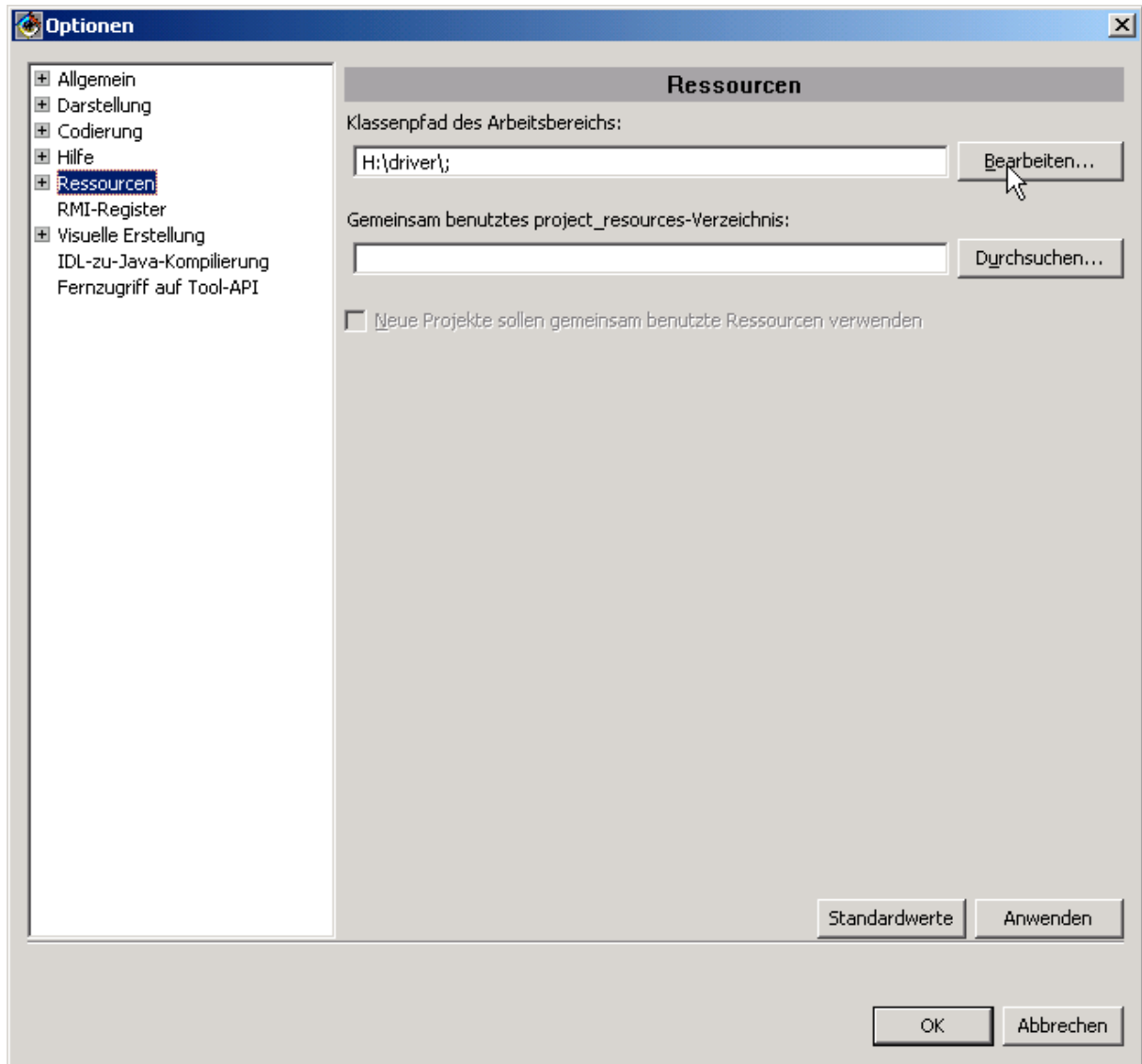
Das Java-Programm "SMJDBCTESTTestPrg" ist für Testzwecke so geschrieben, dass es auf eine OS/390 DB2-Tabelle zugreift, die immer vorhanden ist. Eine solche Tabelle (und die wird hier ausgelesen) ist die DB2-Systemtabelle. Sie enthält einen Eintrag für jede DB2-Tabelle, die angelegt wurde. Im vorliegenden Fall sind das Tabellen, die in den Übungen mit Tutorial 4 unter Benutzung von SPUFI angelegt wurden. Diese Information wird in dem Ausgabe-Fenster wiedergegeben.

Die Java-Klasse "SMJDBCTESTTestPrg" wird auf dem Windows 2000-Rechner ausgeführt. Sie greift über den HIT Driver/Connector auf die DB2-Systemtabelle zu. Die Java-Klasse enthält hierfür die URL, den Namen des Hit Drivers, ein SQL SELECT für die Systemtabelle, plus den Benutzernamen (spruth) und dessen Passwort.

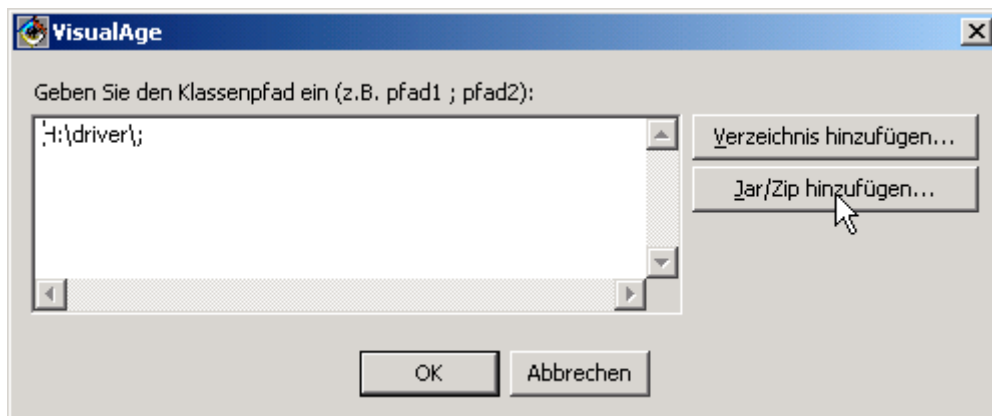
Als nächstes testen wir die Option 3, mit deren Hilfe ein Zugriff auf die gleiche OS/390-Systemtabelle, diesmal aber mit dem DB2Connect-Treiber erfolgt. Damit dies geschehen kann, muss der Treiber noch in dem Klassenpfad hinzugefügt werden.



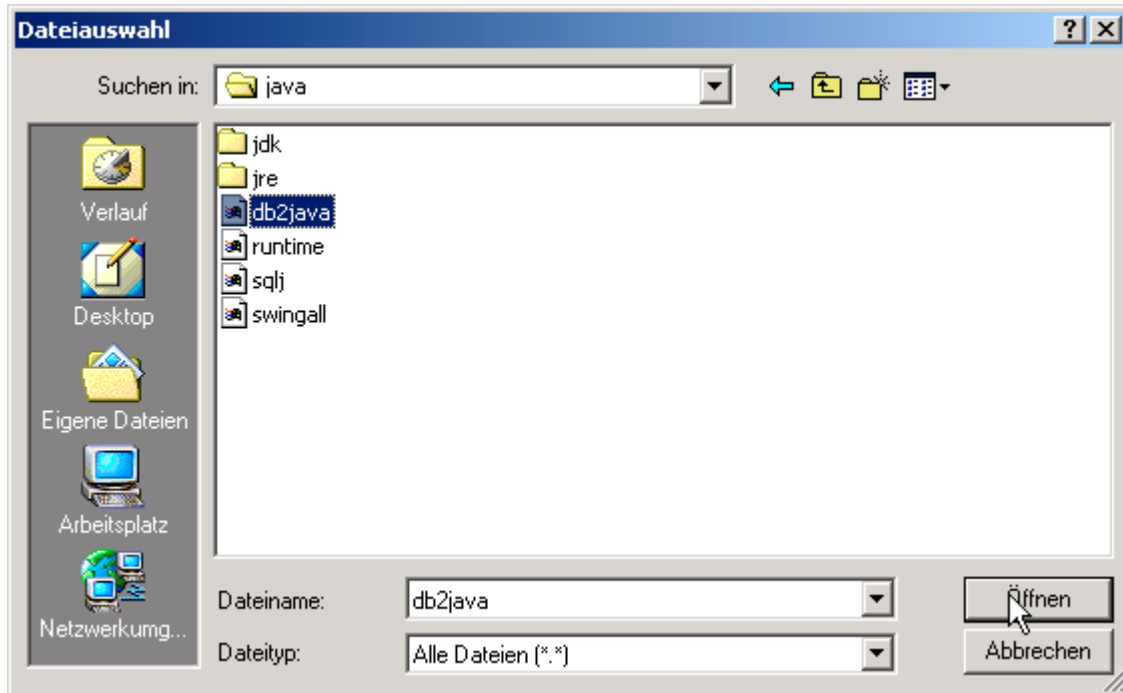
Im Workbench-Fenster ist in der Menüleiste "Fenster" und anschließend "Optionen..." auszuwählen (s. obenstehende Abbildung).



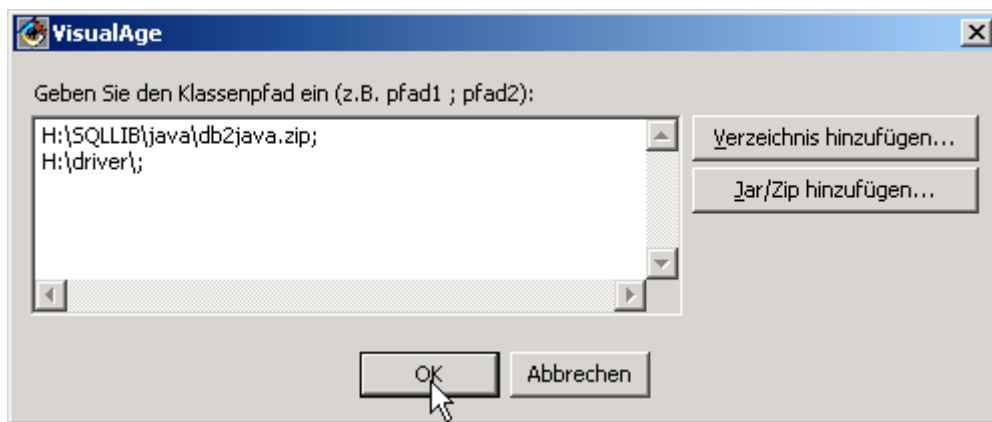
Klicken Sie in obigem Fenster auf den Button "Bearbeiten...".



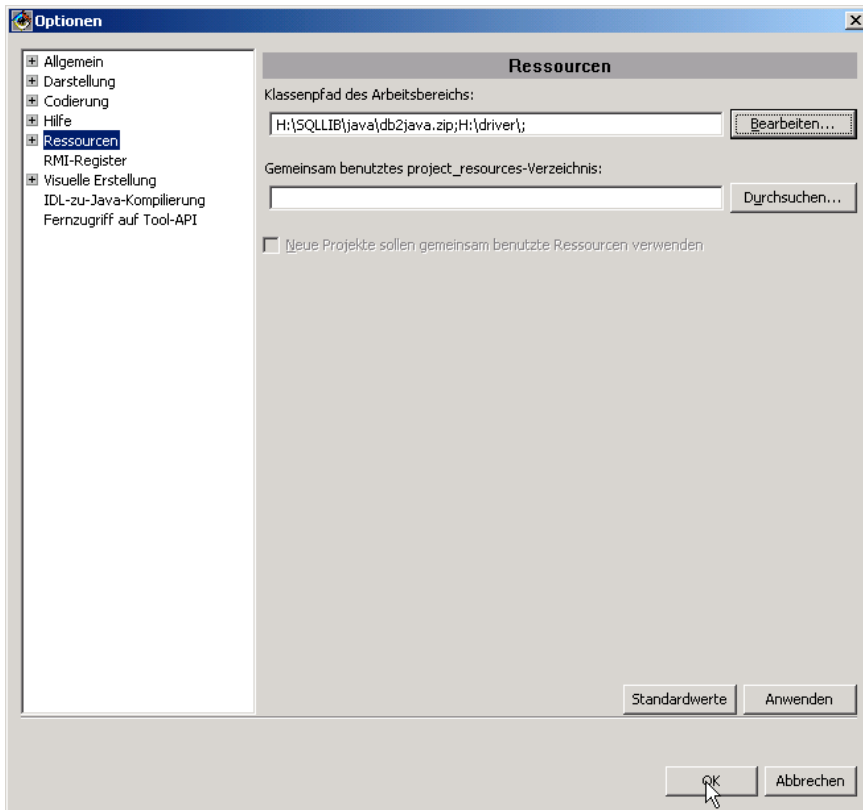
Da sich der erforderliche Treiber in einer Zip-Datei befindet, ist der Button "Jar/Zip hinzufügen..." anzuklicken.



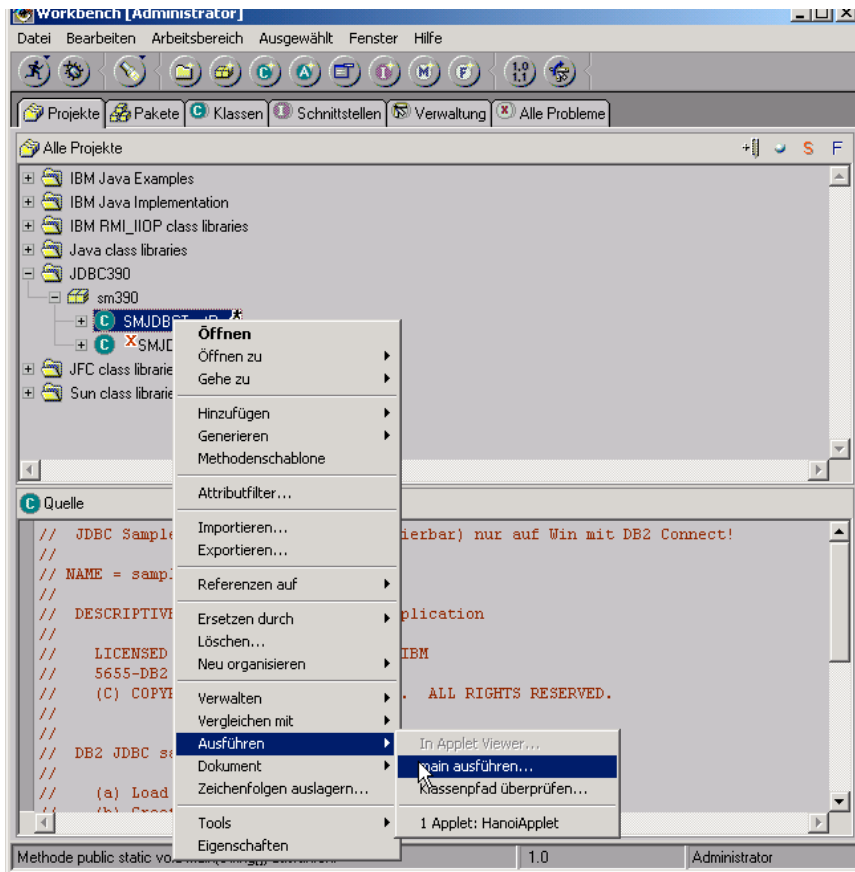
Fügen Sie "H:\SQLLib\java\db2java.zip" durch Klick auf "Öffnen" hinzu.

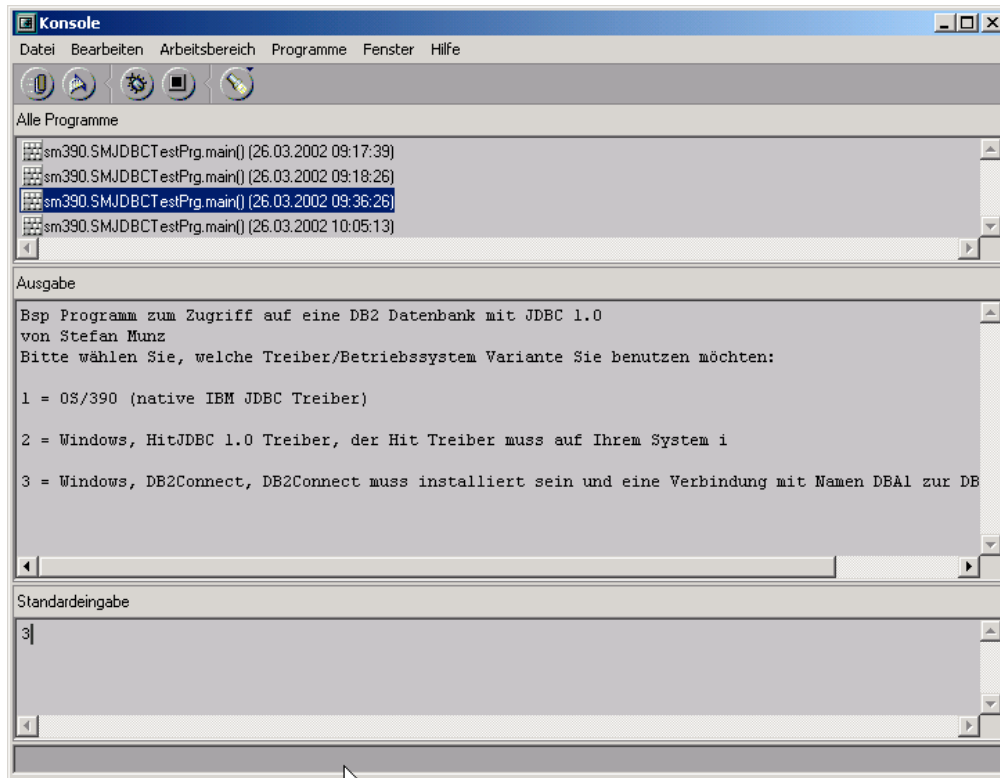


Betätigen Sie den Button "OK".

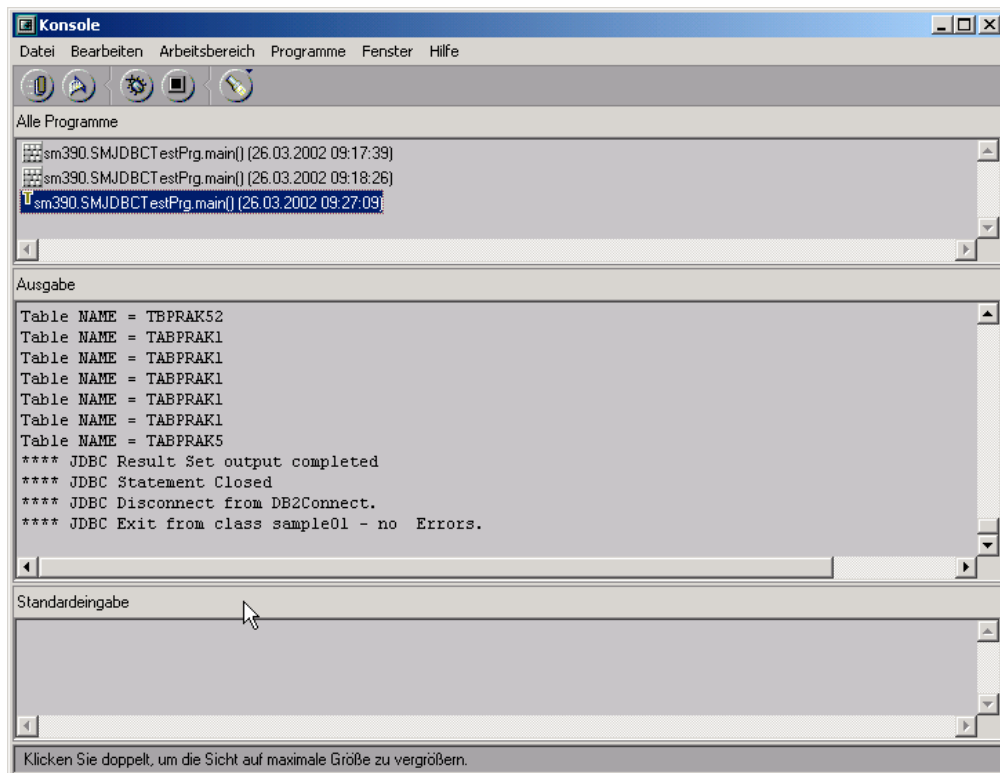


Im Klassenpfad erscheint jetzt "H:\SQLLib\java\db2java.zip", wie obige Abbildung zeigt. Klicken Sie abschließend auf "OK".



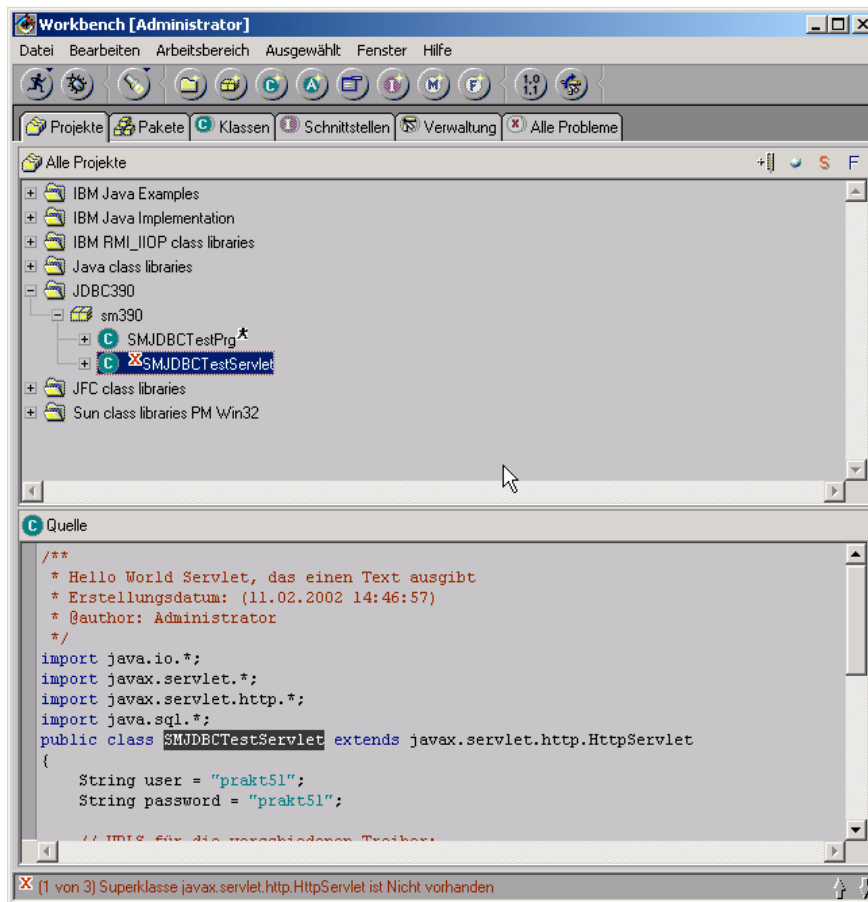


Diesmal die "3" auswählen, um den OS/390 DB2-Zugriff mit Hilfe von DB2Connect durchzuführen.

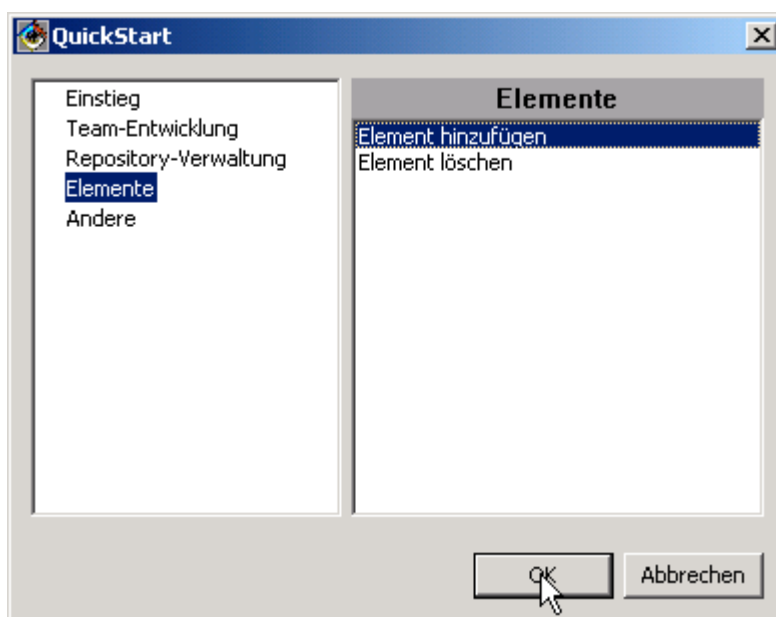


Der OS/390 DB2-Zugriff mit Hilfe von DB2Connect hat das gleiche Ergebnis wie zuvor: Die DB2-Tabellen auf Jedi werden ausgegeben.

## Servlet zum Laufen bringen

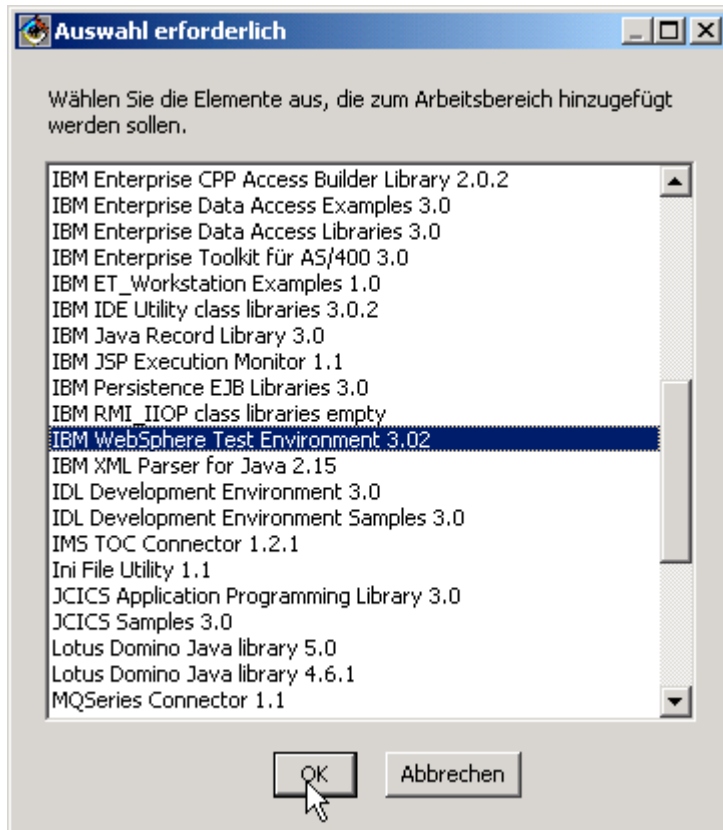


public class SMJDBCTestServlet extends javax.servlet.http.HttpServlet ist nicht vorhanden. VisualAge for Java beinhaltet eine WebSphere-Testumgebung. Letztere beinhaltet wiederum eine Servlet Engine mit der javax.servlet.http.HttpServlet-Klasse. Wir müssen in unsere Workbench hierzu die WebSphere-Testumgebung integrieren.

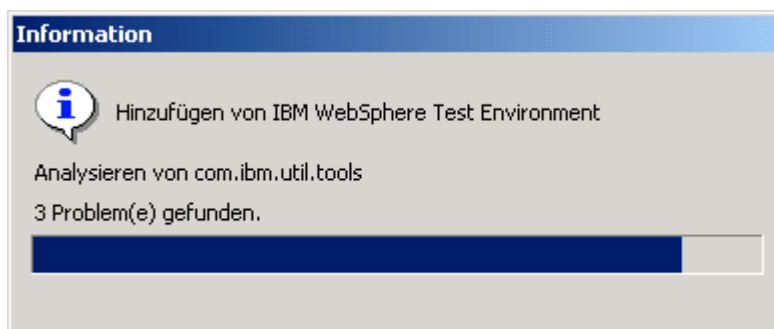




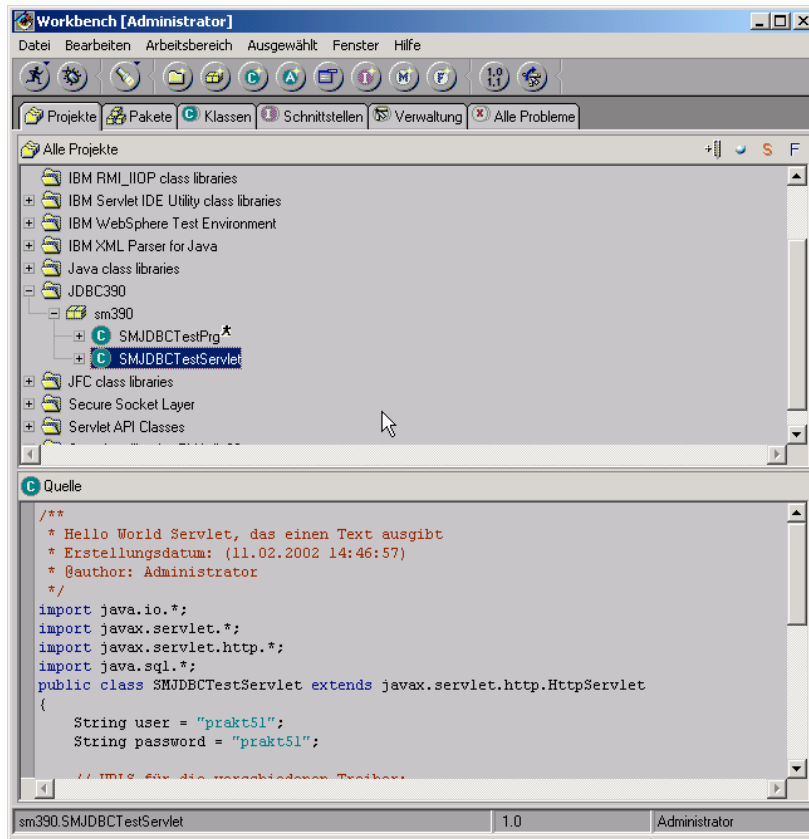
Hierzu im linken Fenster "Elemente", im rechten Fenster "Element hinzufügen" auswählen und anschließend "OK" betätigen.



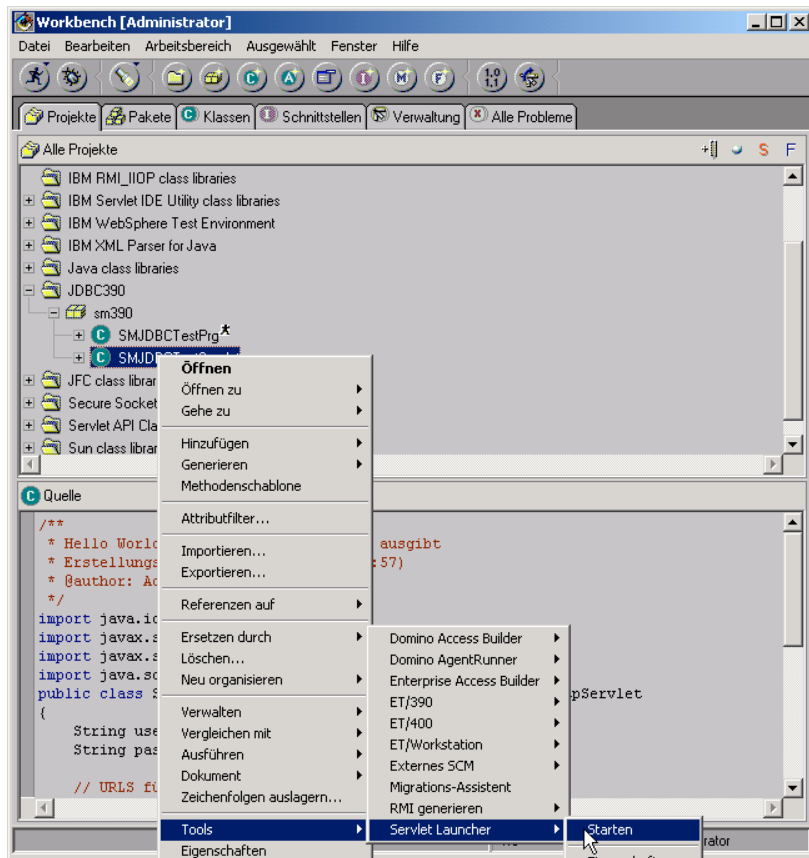
Mit der Scroll Bar die "IBM WebSphere Test Environment 3.02" suchen und markieren, abschließend Klick auf "OK".



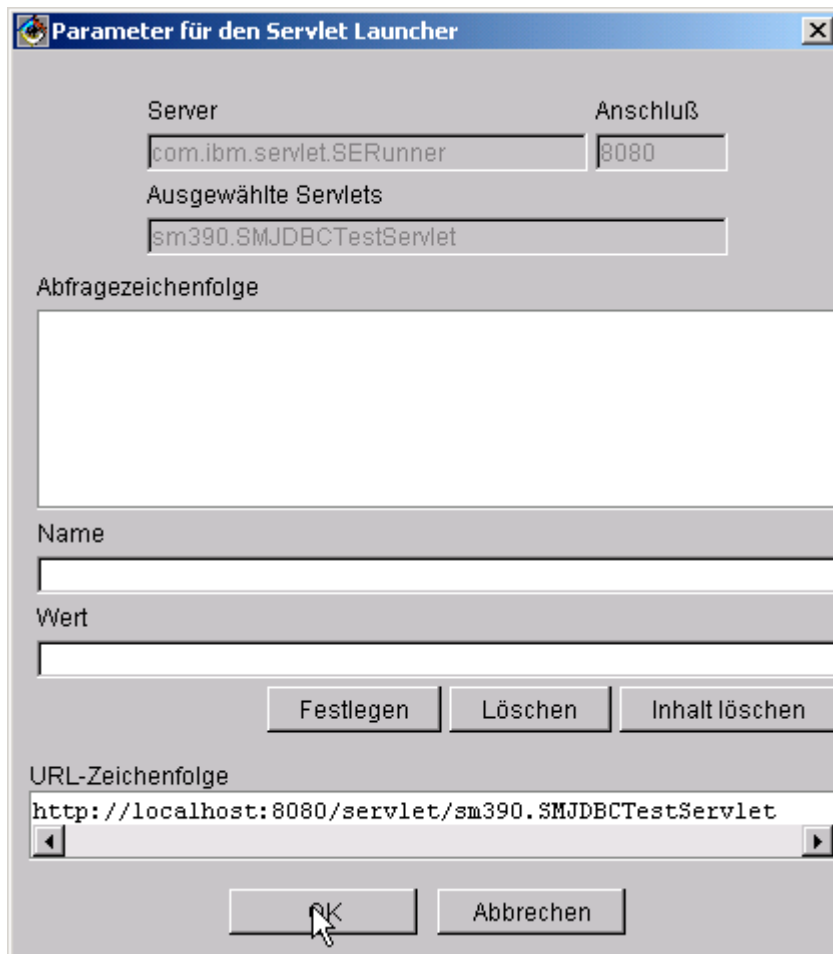
Beim Hinzufügen der Testumgebung werden Probleme gefunden, aber automatisch aufgelöst

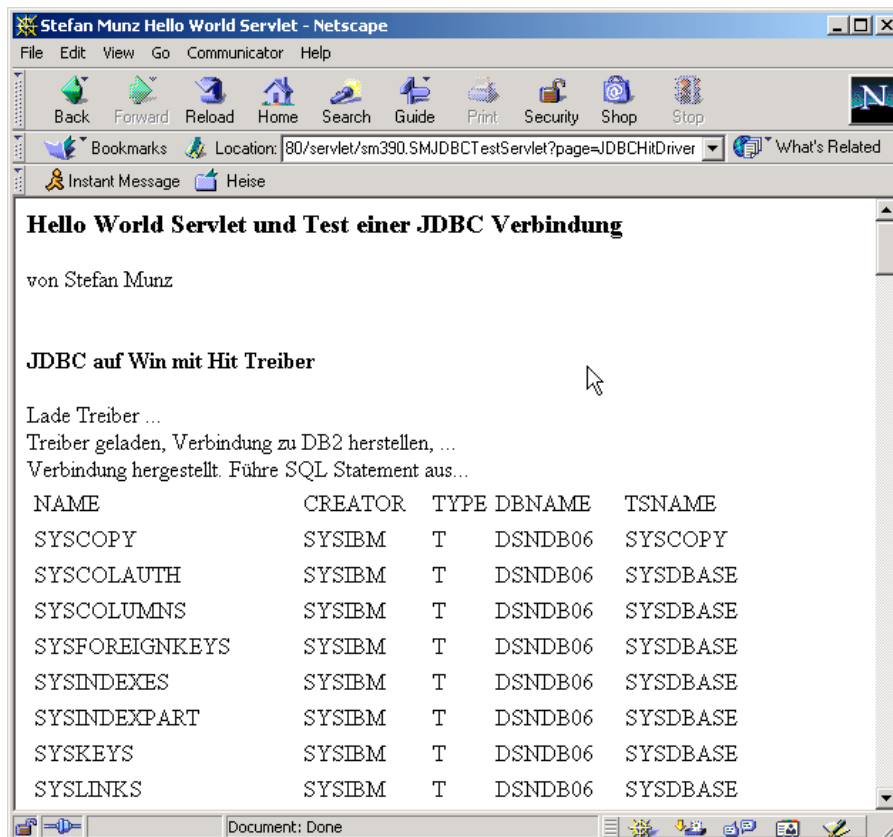


Bei dem "SMJDBCTestServlet" ist jetzt das rote Kreuz <sup>x</sup> verschwunden.



Wir starten das Servlet in einer Testumgebung. Deshalb selektieren wir - wie in obiger Abbildung dargestellt - "Tools", wählen anschließend den "Servlet Launcher" und abschließend "Starten".





Wenn wir statt dessen auf JDBCDB2Connect klicken, erhalten wir die gleiche Ausgabe. Die folgende Aufgabe baut auf die Aufgaben des Tutorials 4 auf.

Aufgabe: Modifizieren Sie das Programm "SMJDBCTestPrg" derart, dass Sie mit diesem auf die DB2-Tabelle zugreifen können, die Sie bei der Bearbeitung des Tutorials 4 angelegt haben. Wie unten dargestellt, sollen die vier Vornamen angezeigt werden, wovon einer der ihre oder bei Gruppen mehrere Vornamen die ihren sein müssen. Der Zugriff kann wahlweise über den Hit-Treiber oder über DB2Connect durchgeführt werden. Schicken Sie ihrem Betreuer einen Screenshot ähnlich dem unten abgebildeten zu. "Table Name" muß natürlich überall durch "Vorname" ersetzt sein. Der Screenshot sowie alle weiteren Screenshots dieses Tutorials müssen im Bitmap- (max. 250 KByte) oder besser noch im JPEG-Format (max. 90 KByte) erstellt sein.

```

Konsole
Datei Bearbeiten Arbeitsbereich Programme Fenster Hilfe
Alle Programme
T sm390.SMJDBCTestPrg.main() [20.05.2003 14:06:45]
T sm390.SMJDBCTestPrg.main() [20.05.2003 14:11:46]

Ausgabe

2 = Windows, HitJDBC 1.0 Treiber, der Hit Treiber muss auf Ihrem System installiert sein
3 = Windows, DB2Connect, DB2Connect muss installiert sein und eine Verbindung mit Namen DBAL zur DB definiert sein

2
**** JDBC Entry within class sample01.
**** JDBC Connection to DB2Connect.
**** JDBC Statement Created
**** JDBC Result Set Created
Vorname = MARTIN
Vorname = FRANZISKA
Vorname = NILS
Vorname = NICOLE
**** JDBC Result Set output completed
**** JDBC Statement Closed
**** JDBC Disconnect from DB2Connect.
**** JDBC Exit from class sample01 - no Errors.

Standardeingabe

```

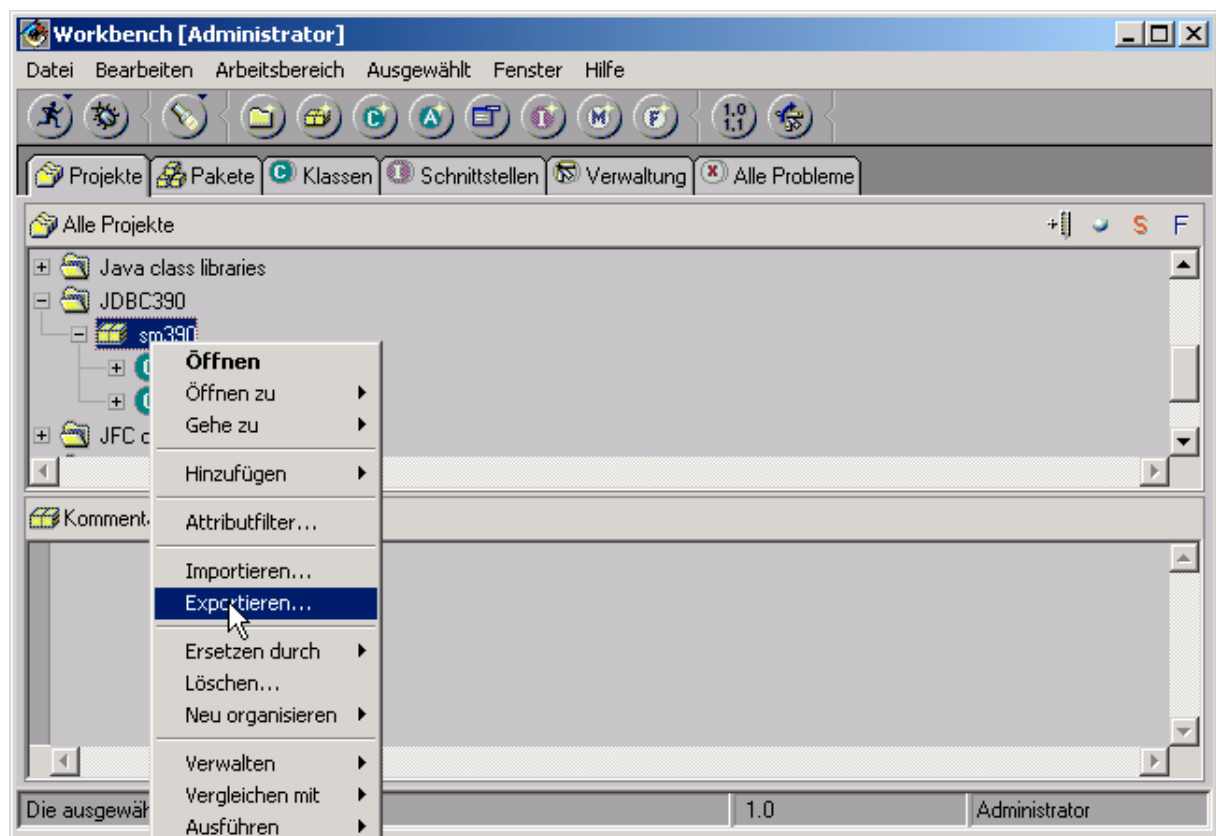
*Aufgabe: Modifizieren Sie jetzt das Servlet "SMJDBCTestServlet" so, dass nun per Servlet auf ihre im Tutorial 4 angelegte Tabelle zugegriffen werden kann. Führen Sie den Zugriff mittels DB2Connect durch, wenn Sie in voriger Aufgabe den Hit-Treiber benutzt haben und umgekehrt. Schicken Sie an ihren Betreuer einen Screenshot ähnlich dem unten abgebildeten.*



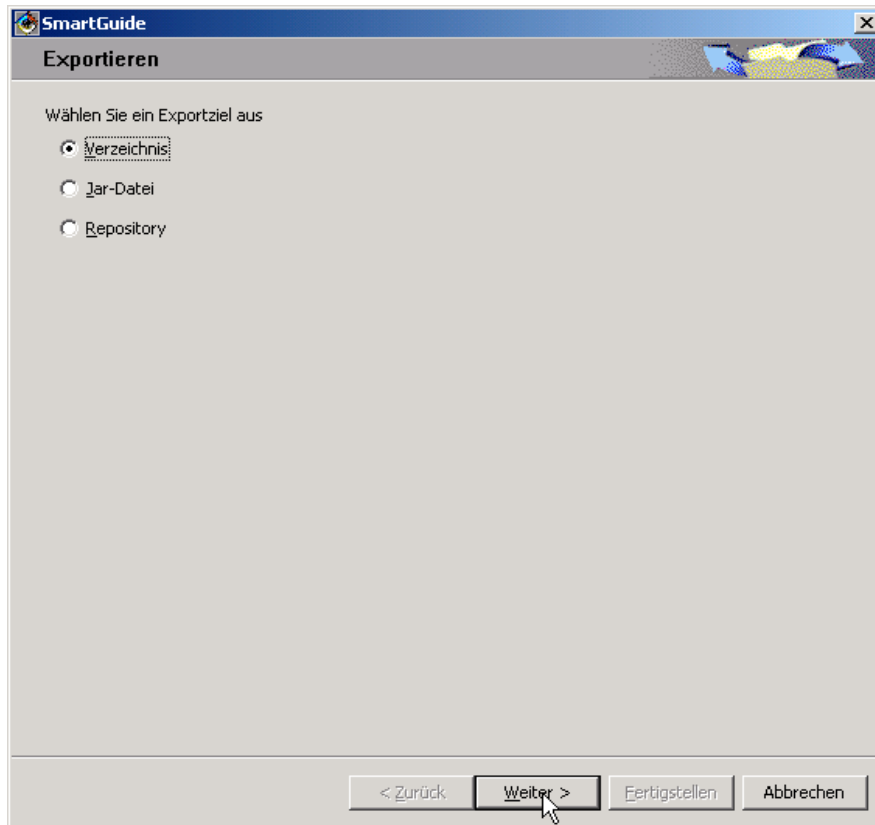
So, das war es. Wir haben demonstriert, wie man von einem Windows 2000-System mit Java-Programmen auf eine OS/390 DB2-Datenbank zugreifen kann.

Die gleichen Java-Programme lassen sich auch auf dem OS/390-Rechner statt auf dem Windows 2000-Rechner ausführen. Hierzu gibt es 2 Möglichkeiten. Wir können die .class Files auf den OS/390-Rechner laden und dort unter Unix System Services direkt ausführen. Alternativ können wir den Java-Quellcode auf den OS/390-Rechner laden, dort übersetzen und dann ausführen.

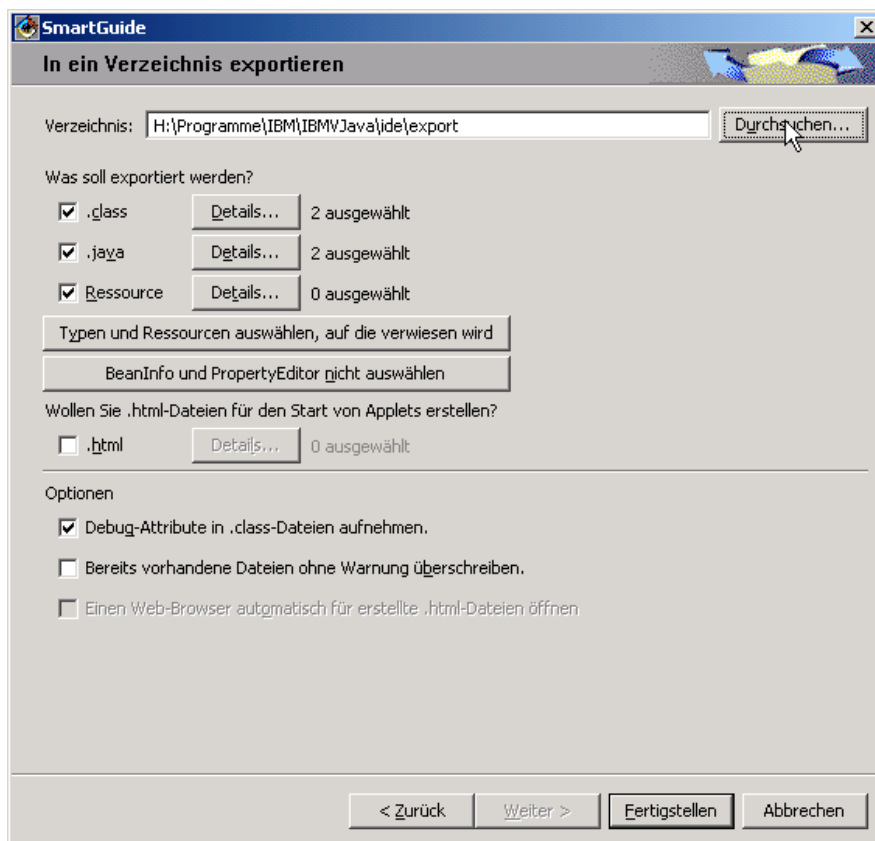
Wir wollen beide Alternativen ausprobieren. Dazu müssen wir die .java und .class Files aus der VisualAge-Entwicklungsumgebung exportieren. Hierfür legen wir zunächst ein Verzeichnis "H:\temp" an und exportieren die Files in dieses Verzeichnis.



In der Workbench rechts auf das zu exportierende Paket "sm390" mit der rechten Maustaste klicken und anschließend "Exportieren" auswählen.

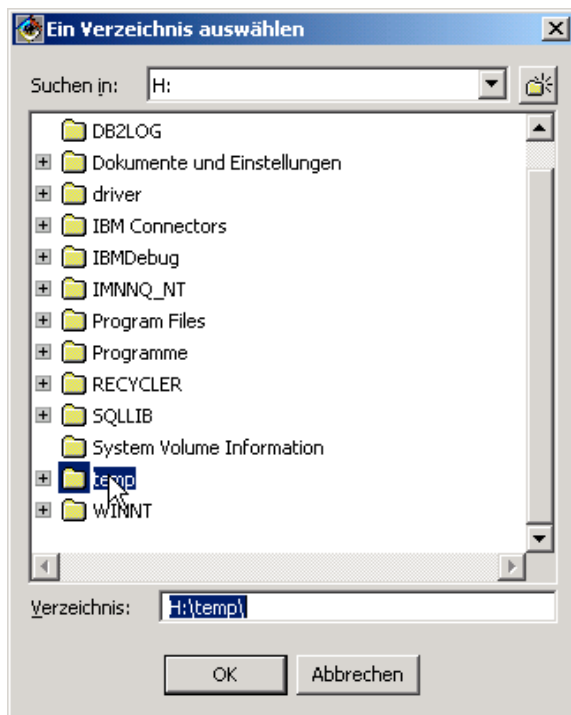


Wir wollen in ein Verzeichnis auf unseren lokalen Plattenspeicher exportieren. Zuvor müssen wir ein Verzeichnis "H:\temp" anlegen, falls dieses noch nicht existiert. Anschließend klicken wir im oben dargestellten "Exportieren"-Fenster auf "Weiter".

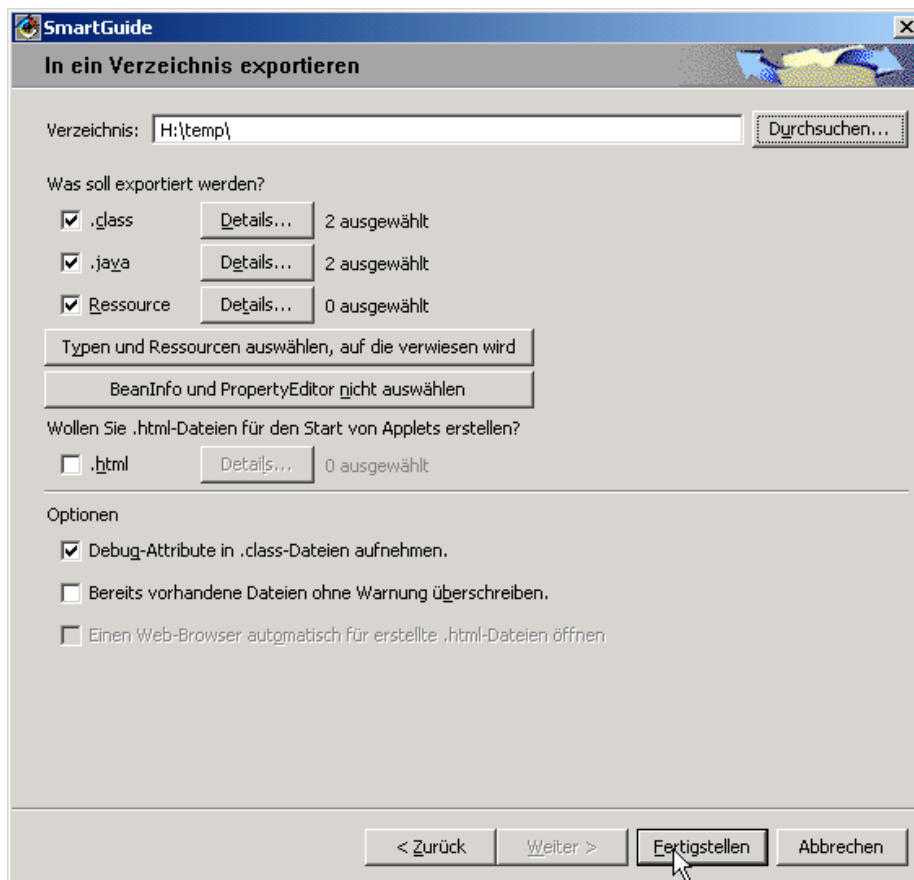




Mittels Durchsuchen selektieren wir unser Zielverzeichnis "H:\temp" ...



und betätigen den Button "OK".



"Fertigstellen" beendet das Exportieren.

Wenn wir jetzt in das "H:\temp"-Verzeichnis schauen, finden wir dort ein Unterverzeichnis "H:\temp\sm390". In diesem Unterverzeichnis finden wir vier Dateien:

```
SMJDBCTestPrg.class  
SMJDBCTestPrg.java  
SMJDBCTestServlet.class  
2SMJDBCTestServlet.java
```

Diese können nun von OS/390 weiterverarbeitet werden.