Tutorial 3

CICS-Anwendung mit JSP als Presentation Logic

Teil 2

JSP als Präsentationslogik von CICS-Anwendungen mit DB/2-Zugriff

Ein sauber strukturiertes CICS-Programm besteht aus zwei Teilen: Anwendung- und der Darstellungslogik. Die Anwendungslogik ist der Teil in dem Berechnungen erfolgen und Daten in einer Datenbank gespeichert bzw. gelesen werden. Die Darstellungslogik ist der Teil, in dem die Ergebnisse der Berechnungen so aufgearbeitet werden, dass Sie dem Benutzer in einer ansprechenden Art auf dem Bildschirm dargestellt werden können.
Für die Darstellungslogik existieren zahlreiche Alternativen. Die Modernste, und auch hier benutze Variante, ist die Verwendung von Java Server Pages, um die Anwendung innerhalb eines Web-Browsers darzustellen. Die Anwendungslogik wird in Sprachen wie C, C++, PL/I, ASSEMBLER usw. geschrieben. Die Verwendung von JSP als Darstellungslogik bedingt die Benutzung von COBOL als Programmiersprache für die Anwendungslogik.
Die Verwndung von JSP impliziert eine 4-Tier-Umgebung aus *„Client“*, *„ServletContainer“*, *„Transaktionsmonitor”*“und *„Datenbank“*. Die Kommunikation zwischen CICS und DB2 ist in den vorhergehenden Tutorials bereits ausführlich erläutert.
Die Kommunikation zwischen Client und Transaktionsmonitor erfolgt über zwei unabhängige Anwendungen. Einerseits der *„ServletContainer“*, der Anfragen vom Client entgegennimmt und andererseits das *„CICS Transaction Gateway“* zur Weiterleitung der Anfrage an den Transaktionsmonitor.
Der CICS Transaction Gateway benutzt die CICS-Schnittstellen *„ECI“* und *„EXCI“* zum Aufruf von Transaktionen. Die Kommunikation zwischen der Java-Anwendung und dem Transaction Gateway erfolgt über ein proprietäres TCP-Protokoll (Port 2006). Die Client-Implementierung ist unter anderem auch als J2CA[1]-Ressource-Adapter verfügbar und kann somit in allen J2EE[2]-konformen Applikationsservern eingesetzt werden.
Die CICS-Schnittstellen *„ECI“* und *„EXCI“* entsprechen dem CICS-API-Aufruf „EXEC CICS LINK“. Dieser API-Aufruf erlaubt es, direkt den Inhalt des Speicherbereiches COMMAREA vor Transaktionsbeginn anzugeben und nach Transaktionsende wieder auszulesen. Die aufgerufene Transaktion ändert den Inhalt dieses gemeinsamen Speicherbereiches und benutzt den CICS-API-Aufruf „EXEC CICS RETURN“, um den geänderten Speicherbereich an die aufrufende Transaktion bzw. das CICS Transaction Gateway zurückzugeben.

1 Java 2 Connector Architecture
2 Java 2 Enterprise Edition

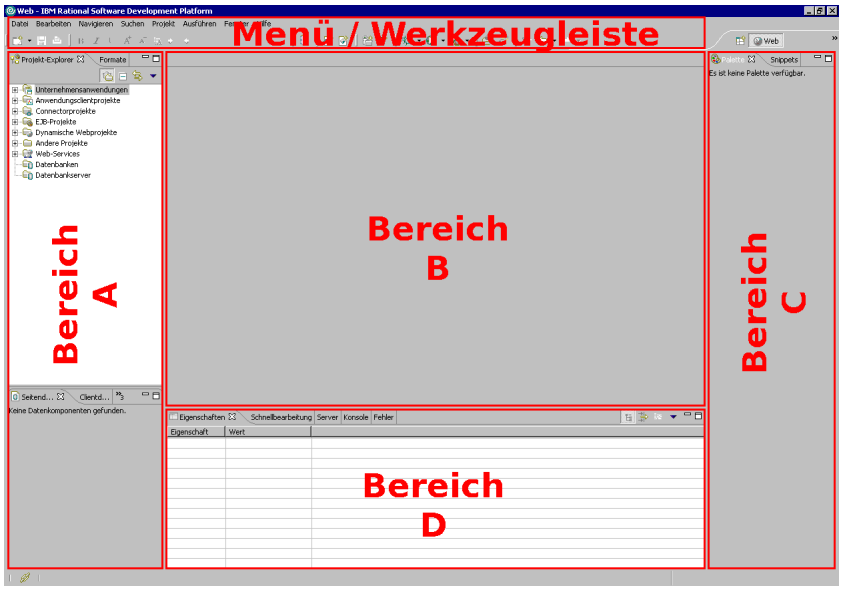

Abbildung 1: Oberfläche der Entwicklungsumgebung

Die Oberfläche der Entwicklungsumgebung *„Websphere Developer für z/Series“* wird in die in Abbildung 1 dargestellten Bereiche unterteilt. Diese Bereiche enthalten verschiedene *„Sichten“*, wie z.B. die Sicht *„Projekt-Explorer“* im Bereich *„Navigator“* der Abbildung 1. Eine bestimmte Kombination und Anordnung von Sichten wird als *„Perspektive“* bezeichnet. Hier ist beispielhaft die Perspektive *„Web“* dargestellt.

Erstellen des Datenbankschemas

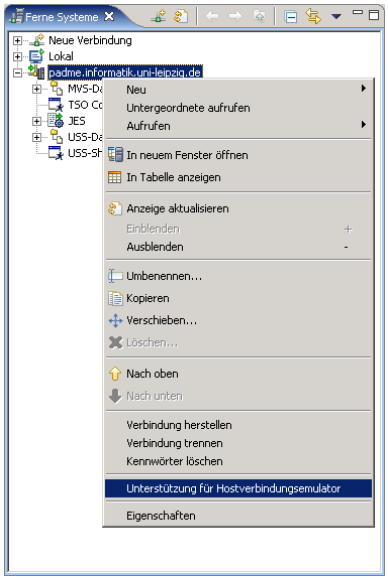

Abbildung 2: Start des Hostverbindungsemulator

Öffnen Sie den integrierten Hostverbindungsemulator, indem Sie mit der rechten Maustaste in der Sicht *„Ferne Systeme“* den Eintrag *„padme.informatik.uni-leipzig.de“* anklicken und im Kontextmenü den Eintrag *„Unterstützung für Hostverbindung“* wählen. (Abbildung 2)

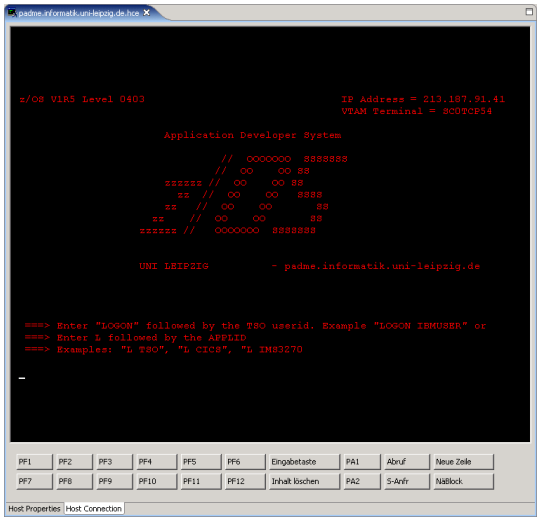

Abbildung 3: Hostverbindungsemulator

Der Emulator wird gestartet und in der Mitte des Arbeitsbereiches angezeigt.

```
                          SPUFI                                SSID: DB7G
===>

Enter the input data set name:        (Can be sequential or partitioned)
 1  DATA SET NAME ... ===> SPUFI.IN(GENSCHMA)
 2  VOLUME SERIAL ... ===>            (Enter if not cataloged)
 3  DATA SET PASSWORD ===>            (Enter if password protected)

Enter the output data set name:       (Must be a sequential data set)
 4  DATA SET NAME ... ===> SPUFI.OUT

Specify processing options:
 5  CHANGE DEFAULTS   ===> YES        (Y/N - Display SPUFI defaults panel?)
 6  EDIT INPUT ...... ===> YES        (Y/N - Enter SQL statements?)
 7  EXECUTE ......... ===> YES        (Y/N - Execute SQL statements?)
 8  AUTOCOMMIT ...... ===> YES        (Y/N - Commit after successful run?)
 9  BROWSE OUTPUT ... ===> YES        (Y/N - Browse output data set?)

For remote SQL processing:
10  CONNECT LOCATION  ===>


 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Abbildung 4: Aufruf von SPUFI

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------------
EDIT       PRAKT20.SPUFI.IN(GENSCHMA) - 01.01              Columns 00001 00072
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000100 CREATE STOGROUP STOGP020 VOLUMES (SMS001) VCAT DSN710;
000200 CREATE DATABASE DBPRK020 STOGROUP STOGP020 BUFFERPOOL BP0;
000300 CREATE TABLESPACE TBSPP020
000400        IN DBPRK020
000500        USING STOGROUP STOGP020
000600              PRIQTY 20
000700              SECQTY 20
000800              ERASE NO
000900        BUFFERPOOL BP0
001000        CLOSE NO;
001100 CREATE TABLE TBLPR020 (
001200        VORNAME VARCHAR(24) NOT NULL,
001300        NAME VARCHAR(24) NOT NULL)
001600        IN DBPRK020.TBSPP020;
001700 INSERT INTO TBLPR020(VORNAME, NAME)
Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Abbildung 5: Erster Teil des Datenbankschemas

Öffnen Sie eine TSO-Sitzung und starten Sie die ISPF-Anwendung *„SPUFI“* zur Erstellung eines neuen Datenbankschemas. Wählen Sie „SPUFI.IN(GENSCHMA)“ als Name für den Eingabe-

Dataset und wählen Sie „SPUFI.OUT“ als Ausgabe-Dataset.
Geben Sie nun die Definition des Datenbankschemas in Form von SQL-Statements ein. Achten Sie darauf, dass alle Statements mit SQL-CODE=0 ausgeführt wurden. Zur Fehleranalyse können die Entsprechungen dieses Rückgabewertes im *„DB/2 Messages and Codes Guide, Kapitel 2“* nachgelesen werden.

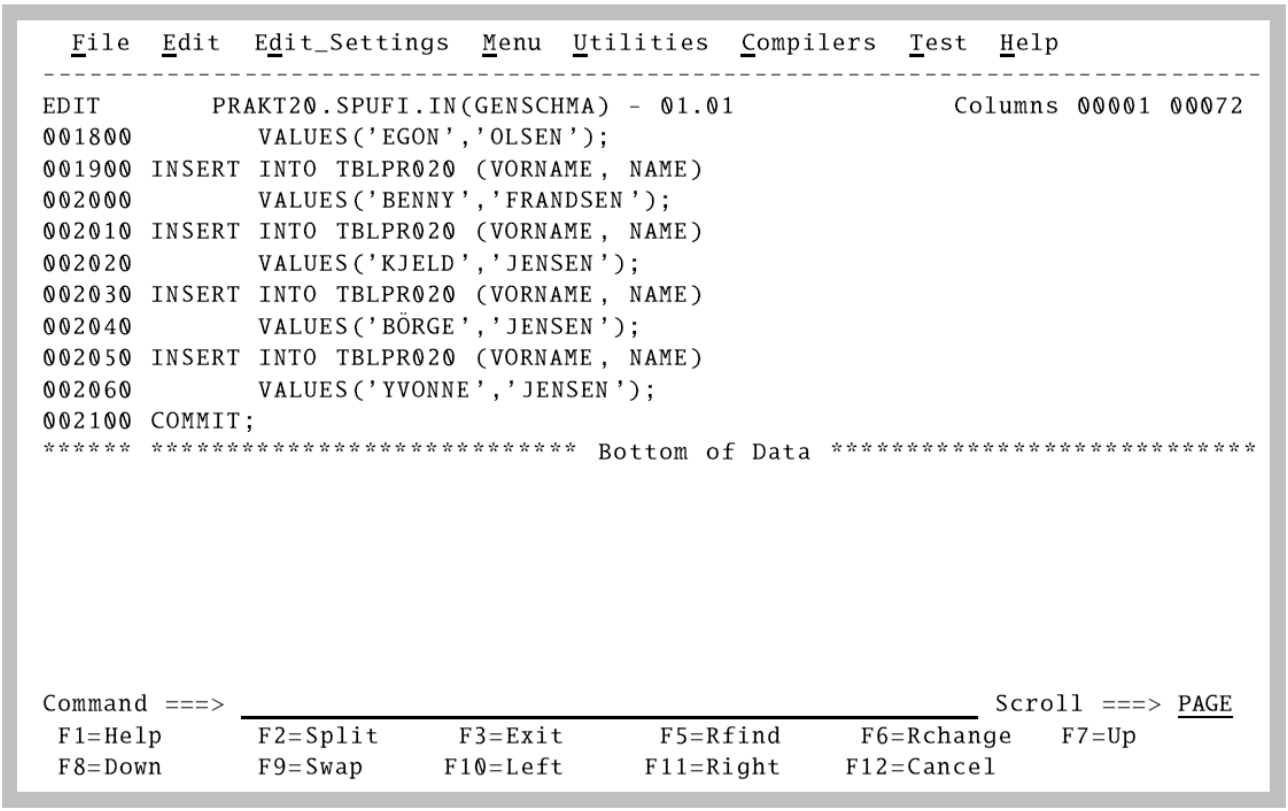

Abbildung 6: Zweiter Teil des Datenbankschemas

Aufgabe:	*Erstellen Sie ein eigenes Datenbankschema bestehend aus einer Tabelle mit einigen wenigen Spalten. Füllen Sie die Tabelle mit einigen Datensätzen. Die Datensätze müssen Ihren Namen enthalten. Wird das Tutorial durch mehrere Personen bearbeitet, dann müssen die Namen aller Bearbeiter enthalten sein.*

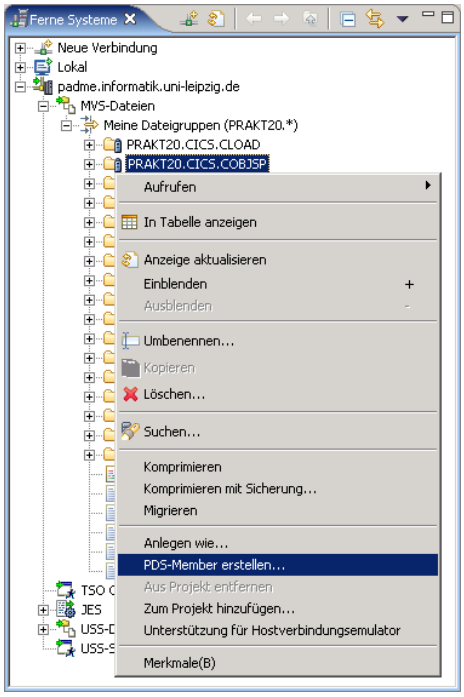

Abbildung 7: PDS-Member erstellen

Erstellen Sie ein neues PDS-Member im Dataset „PRAKT20.CICS.COBJSP“, indem Sie mit der rechten Maustaste auf den Dataset klicken und im Kontextmenü den Eintrag *„PDS-Member erstellen“* auswählen.

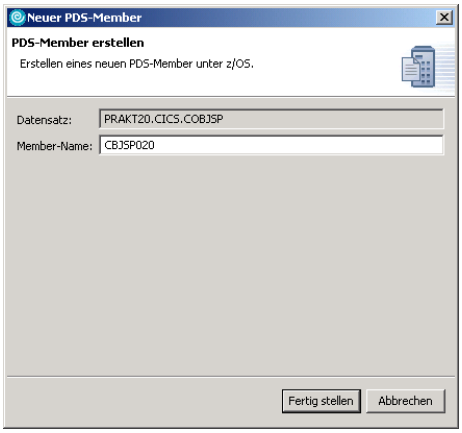

Abbildung 8: Name für neues PDS-Member

Geben Sie im daraufhin geöffneten Dialog (Abbildung 8) als Member-Name „CBJSP020“ an und bestätigen Sie ihre Eingabe mit der Taste *„Fertig stellen“*.

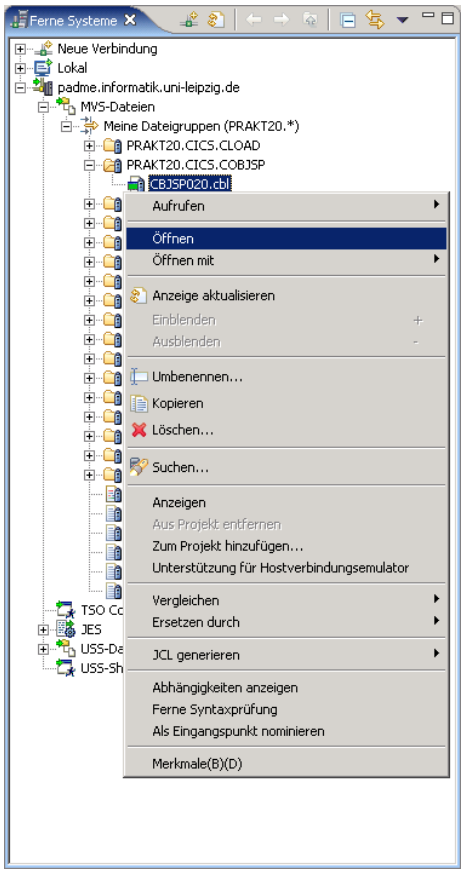

Abbildung 9: CBJSP020 öffnen

Erweitern Sie den Eintrag „PRAKT20.CICS.COBJSP“ in der Sicht *„Ferne Systeme“* durch Klick auf das Symbol (+) vor dem Eintrag. Öffnen Sie das Member durch Klick mit der rechten Maustaste auf den Eintrag „CBJSP020.cbl“ und die Auswahl von *„Öffnen“* im Kontextmenü.

Übertragen Sie den Quelltext für das Member entsprechend der Abbildung 10.
Die Anwendung ermittelt zunächst die Anzahl der in der Tabelle enthaltenen Datensätze und speichert diese Information in der Variable „ANZAHL“. Es werden maximal die ersten 16 Datensätze betrachtet. Mit Hilfe eines Cursors werden die Datensätze aus der Datenbank nacheinander in der Datenstruktur „NAMEN“ gespeichert. Durch den CICS-API-Auftruf „EXEC CICS RETURN“ wird der gesamte Speicherbereich „DFHCOMMAREA“ dem aufrufenden Programm (in diesem Fall das CICS-Transaction-Gateway) zur Verfügung gestellt.

```
 1      IDENTIFICATION DIVISION.
 2      PROGRAM-ID. CBJSP020.
 3      ENVIRONMENT DIVISION.
 4      DATA DIVISION.
 5      WORKING-STORAGE SECTION.
 6          EXEC SQL INCLUDE SQLCA END-EXEC.
 7      01 RCOUNT       PIC S9(9)        COMP-3.
 8      01 RCOUNTER     PIC 99                    VALUE 1.
 9      01 N1           PIC X(24).
10      01 N2           PIC X(24).
11      LINKAGE SECTION.
12      01 DFHCOMMAREA.
13         02 ANZAHL    PIC 99.
14         02 FEHLER    PIC S9(9).
15         02 FEHLERTXT PIC X(70).
16         02 NAMEN     OCCURS 16 TIMES.
17            03 NAME1  PIC X(24).
18            03 NAME2  PIC X(24).
19     ****************************************************************
20      PROCEDURE DIVISION.
21      MAIN.
22          MOVE 0 TO ANZAHL.
23          MOVE 0 TO FEHLER.
24     ****************************************************************
25          EXEC SQL
26              SELECT COUNT(*) INTO :RCOUNT FROM TBLPR020
27          END-EXEC.
28          IF SQLCODE LESS THAN 0 THEN
29              PERFORM DISPLAY-ERROR
30          END-IF.
31          IF RCOUNT GREATER THAN 16 THEN MOVE 16 TO RCOUNT END-IF.
32          MOVE RCOUNT TO ANZAHL.
33     ****************************************************************
34          EXEC SQL
35              DECLARE C1 CURSOR FOR
36              SELECT VORNAME,NAME FROM TBLPR020
37              FETCH FIRST 16 ROWS ONLY
38          END-EXEC.
39          IF SQLCODE LESS THAN 0 THEN
40              PERFORM DISPLAY-ERROR
41          END-IF.
42          EXEC SQL OPEN C1 END-EXEC.
43          IF SQLCODE LESS THAN 0 THEN
44              PERFORM DISPLAY-ERROR
45          END-IF.
46          PERFORM RCOUNT TIMES
47              EXEC SQL
48                  FETCH C1 INTO :N1, :N2
49              END-EXEC
50              IF SQLCODE LESS THAN 0 THEN
51                  PERFORM DISPLAY-ERROR
52              END-IF
53              MOVE N1 TO NAME1(RCOUNTER)
54              MOVE N2 TO NAME2(RCOUNTER)
55              COMPUTE RCOUNTER = RCOUNTER + 1
56          END-PERFORM.
57          EXEC SQL CLOSE C1 END-EXEC.
58          EXEC CICS RETURN END-EXEC.
59          EXIT.
60     ****************************************************************
61      DISPLAY-ERROR.
62          MOVE 0 TO ANZAHL.
63          MOVE SQLCODE TO FEHLER.
64          MOVE SQLERRMC to FEHLERTXT.
65          EXEC CICS RETURN END-EXEC.
66          EXIT.
```

Abbildung 10: Anwendung CBJSP020

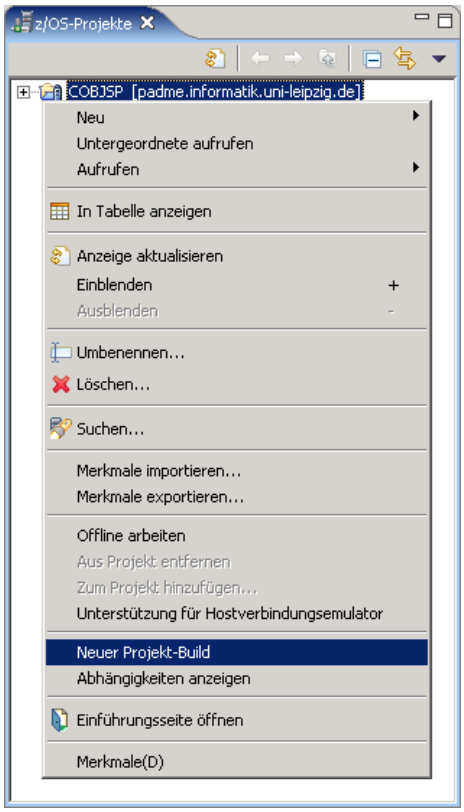

Abbildung 11: MVS-Projekt übersetzen

Um die Anwendung zu übersetzen, klicken Sie mit der rechten Maustaste auf das Projekt *„COBJSPMVS“* in der Sicht *„z/OS-Projekte“* und wählen im Kontextmenü den Eintrag *„Neuer Projekt Build“* (s. Abbildung 11). Die Statusausgabe erfolgt analog den Abbildungen 12 und 13.

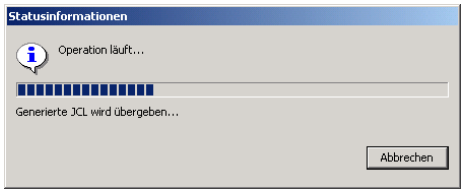

Abbildung 12: JCL generieren und übergeben

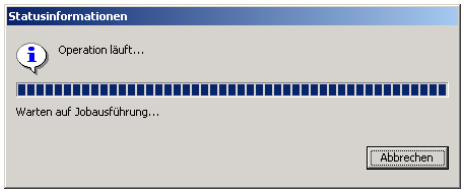

Abbildung 13: Übersetzung abwarten und Ergebnis analysieren

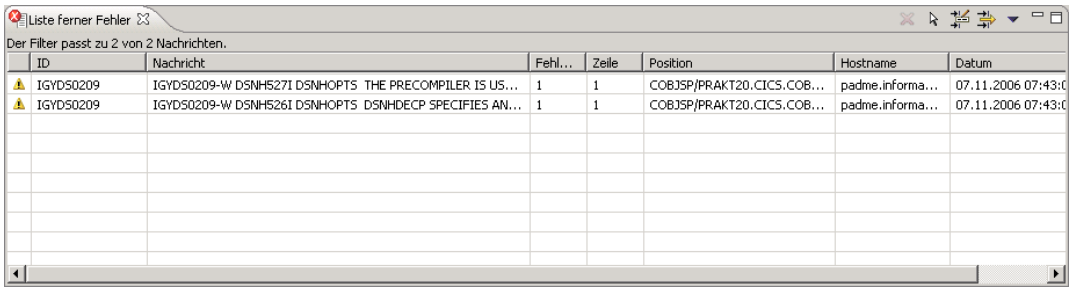

Abbildung 14: Liste ferner Fehler

Die Fehlerausgabe des Übersetzungsvorgangs erscheint in der Sicht *„Liste ferner Fehler“*. Die beiden in Abbildung 14 dargestellten Warnungen mit der ID IGYDS0209 können ignoriert werden.

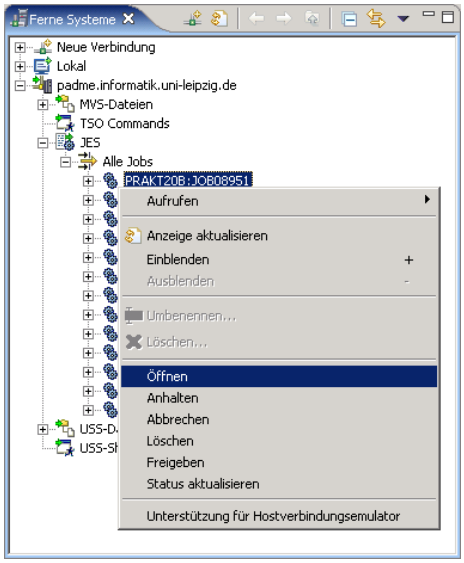

Abbildung 15: JES Job-Log

Die komplette Ausgabe des Build-Scriptes kann, wie in Abbildung 15 dargestellt, über das Job-Log in der Sicht *„Ferne Systeme“* angezeigt werden. Klicken Sie dazu mit der rechten Maustaste auf den gewünschten Job und wählen Sie im Kontextmenü den Eintrag *„Öffnen“*.

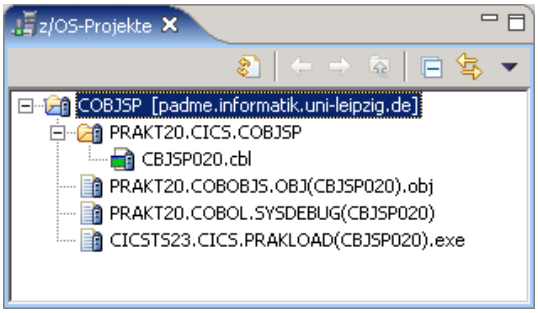

Abbildung 16: Projektstruktur nach erfolgreichem Build

Nach erfolgreicher Übersetzung hat Ihr MVS-Projekt die in Abbildung 16 dargestellte Struktur. Damit ist die Entwicklung der Anwendungslogik abgeschlossen.

Aufgabe:	*Erstellen Sie ein eigenes, zu Ihrem Datenbankschema passendes COBOL-Programm Die Ausgabe Ihrer Anwendung muss unbedingt Ihren Namen enthalten. Ersetzen Sie in den Bezeichnern für Ihr COBOL-Programm und Ihr JCL-Script die Ziffern 020 durch die Nummer Ihres Praktxx bzw. Prakxxx-Account.*

Aufgabe:	*Kopieren Sie die Datei CBJSP020 aus ihrem MVS-Projekt in eine lokale Datei indem Sie die Datei mit der rechten Maustaste in der Sicht* „z/OS-Projekte“ *anklicken und im Kontextmenü den Eintrag Kopieren wählen.* *Öffnen Sie dann in der glichen Sicht den Baum Lokale Dateien und speichern Sie die Datei unter „C:\Dok. u. Einstellungen … Eigene Dateien“. Sie benötigen die Datei später um die COMMAREA-Datenstruktur einzulesen.*

Installation der COBOL-Anwendung in CICS

Die Anwendung muss nun in CICS installiert werden. Dazu öffnen Sie wieder den Hostverbindungsemulator wie in den Abbildungen 2 und 3 dargestellt. Starten Sie eine CICS-Session und legen mit Hilfe des Befehls *„CEDA DEFINE“* ein neues Programm, eine neue Transaktion und einen DB2-Entry an.

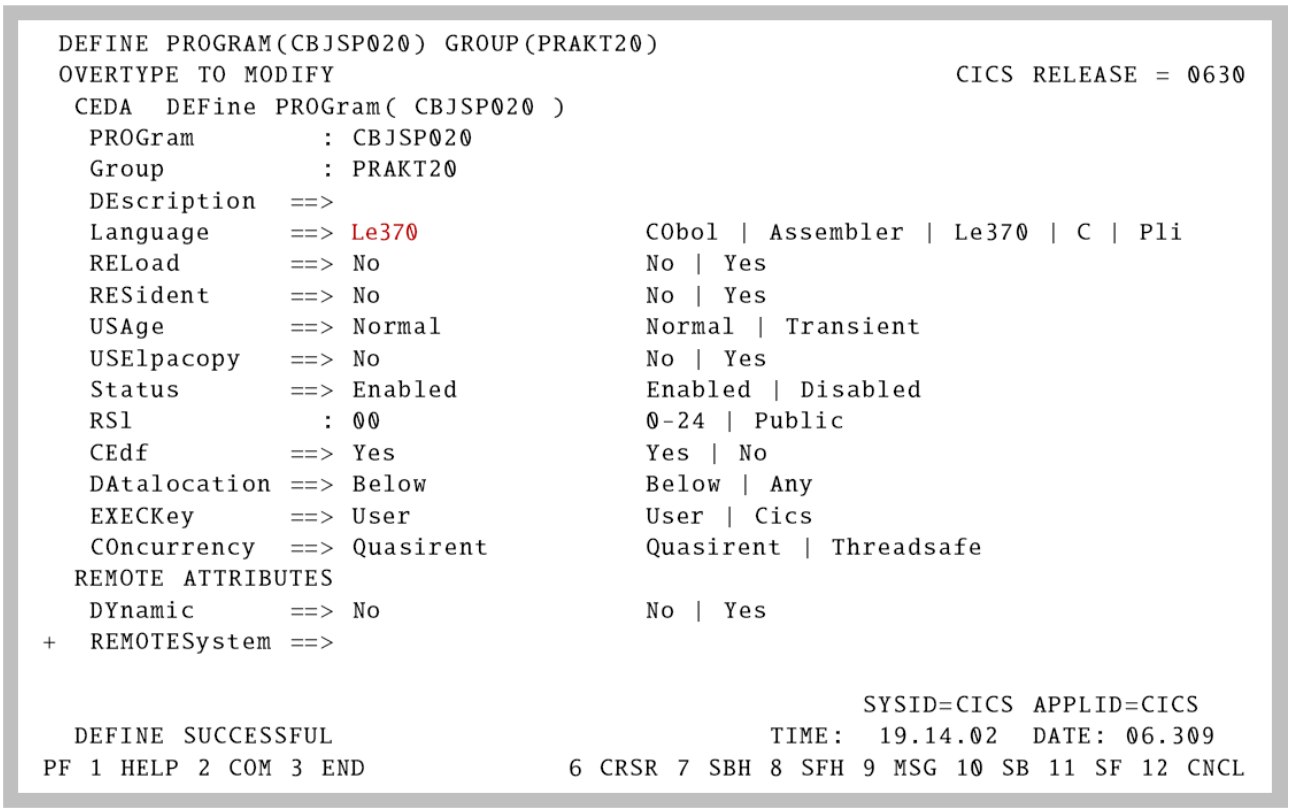

Abbildung 17**:** Definition des CICS-Programms

Legen Sie zunächst mit Hilfe von „CEDA DEFINE PROGRAM(CBJSP020) GROUP(PRAKT20)“ ein neues Programm an. Wichtig ist hierbei die Angabe des Language Environment. Hier ist der Wert „Le370“ einzutragen.

```
 DEFINE TRANSACTION(J020) GROUP(PRAKT20)
 OVERTYPE TO MODIFY                                          CICS RELEASE = 0630
  CEDA  DEFine TRANSaction( J020 )
   TRANSaction  ==> J020
   Group        ==> PRAKT20
   DEscription  ==>
   PROGram      ==> CBJSP020
   TWasize      ==> 00000              0-32767
   PROFile      ==> DFHCICST
   PArtitionset ==>
   STAtus       ==> Enabled            Enabled | Disabled
   PRIMedsize     : 00000              0-65520
   TASKDATALoc  ==> Below              Below | Any
   TASKDATAKey  ==> User               User | Cics
   STOrageclear ==> No                 No | Yes
   RUnaway      ==> System             System | 0 | 500-2700000
   SHutdown     ==> Disabled           Disabled | Enabled
   ISolate      ==> Yes                Yes | No
   Brexit       ==>
+ REMOTE ATTRIBUTES
  S PROGRAM OR REMOTESYSTEM MUST BE SPECIFIED.
                                                      SYSID=CICS APPLID=CICS

PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

Abbildung 18: Definition der CICS-Transaktion

Danach legen Sie mit Hilfe von „CEDA DEFINE TRANSACTION(J020) GROUP(PRAKT20)“ eine neue Transaktion mit dem Namen „J020“ an. Wichtig ist hierbei die Angabe des in CICS registrierten Programmnamens.

```
 DEFINE DB2ENTRY(CJDBE020) GROUP(PRAKT20)
 OVERTYPE TO MODIFY                                        CICS RELEASE = 0630
  CEDA  DEFine DB2Entry( CJDBE020 )
   DB2Entry       : CJDBE020
   Group          : PRAKT20
   DEscription  ==>
  THREAD SELECTION ATTRIBUTES
   TRansid      ==> J020
  THREAD OPERATION ATTRIBUTES
   ACcountrec   ==> None              None | TXid | TAsk | Uow
   AUTHId       ==>
   AUTHType     ==> Userid            Userid | Opid | Group | Sign | TErm
                                      | TX
   DRollback    ==> Yes               Yes | No
   PLAN         ==> CBJPL020
   PLANExitname ==> DSNCUEXT
   PRIority     ==> High              High | Equal | Low
   PROtectnum   ==> 0000              0-2000
   THREADLimit  ==> 0003              0-2000
   THREADWait   ==> Yes               Pool | Yes | No

                                                  SYSID=CICS APPLID=CICS
  DEFINE SUCCESSFUL                             TIME:  19.15.45  DATE: 06.309
PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

Abbildung 16: Definition des DB2-Entry

Die letzte Komponente der Gruppe ist der DB2-Entry. Er dient dazu einen Datenbankzugriffs-Thread auszuwählen und die Kommunikation mit der Datenbank sicherzustellen. Tragen Sie hier folgende Werte ein :

Transid	J020
PLAN	CBJPL020
THREADLimit	0003
THREADWait	Yes

Aufgabe:

Installieren Sie ihre Anwendung in CICS.
Achten Sie darauf, dass Sie den DB2-Entry beim ersten mal richtig erstellen Sie können ihn nach einmaliger Installation aus Berechtigungsgründen nicht mehr verändern. Die CICS-Gruppe wird danach nur noch partiell installiert, d.h. Änderungen an der Anwendungslogik und erneute Installation des Programms sind möglich, jedoch keine Änderung des DB2-Plan oder der Thread-Parameter des DB2-Entry! Ersetzen Sie in allen einzugebenden Bezeichnern die Ziffernfolge 020 durch die Nummer Ihres Praktxx bzw. Prakxxx-Account.

Erstellen der Darstellungsslogik

Die Darstellungslogik wird in einem separaten Projekt entwickelt. Benutzen Sie das Menü „*Datei → Neu → Andere*“, um ein neues Projekt zu erstellen.

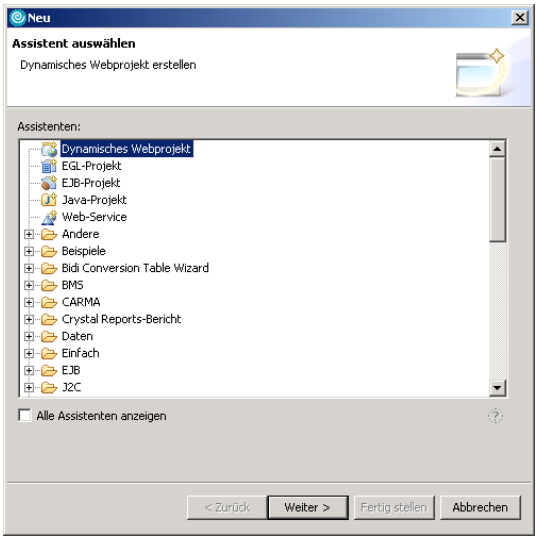

Abbildung 20**:** Neues dynamisches Webprojekt erstellen

Wählen Sie dann, wie in Abbildung 20 dargestellt, den Eintrag *„Dynamisches Webprojekt“* und bestätigen Sie die Auswahl mit der Taste *„Weiter“*.

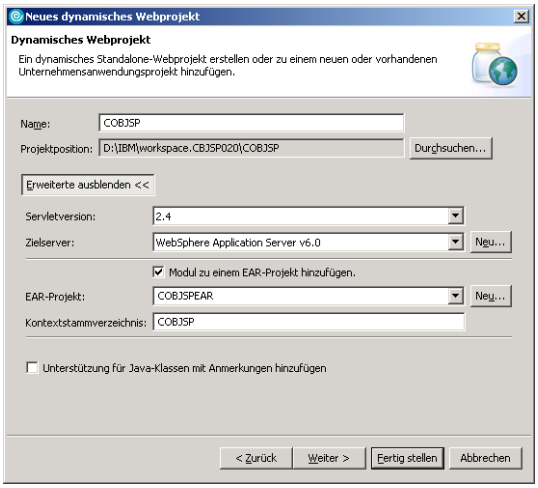

Abbildung 21**:** Neues dynamisches Webprojekt erstellen

Der Dialog *„Neues dynamisches Webprojekt“* wird geöffnet. Geben Sie als Projektname „COBJSP“ an. Wählen Sie die Taste *„Erweiterte anzeigen“* und vergewissern Sie Sich, daß alle Felder bis auf *„Projektposition“* und *„Name“* mit denen in Abbildung 20 übereinstimmen. Schließen Sie danach den Dialog mit der Taste *„Fertig stellen“*.

Abbildung 22: In Perspektive Web wechseln

Das neue Projekt wird jetzt erstellt. Danach werden Sie, wie in Abbildung 22 dargestellt, gefragt, ob Sie in die Perspektive *„Web“* wechseln möchten. Beantworten Sie die Frage durch Betätigen der Taste *„Ja“*.

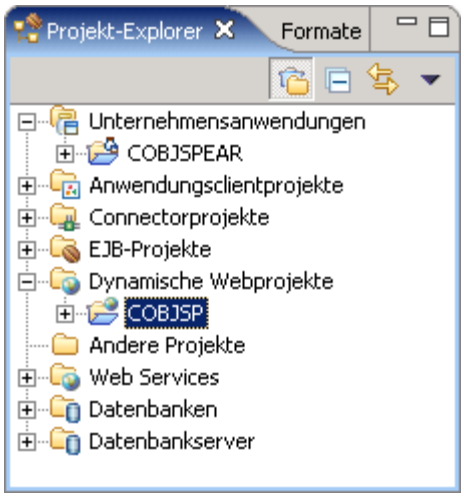

Abbildung 23: Projektexplorer in Perspektive Web

Am linken Bildschirmrand wird jetzt die Sicht *„Projekt-Explorer“* analog zu Abbildung 23 angezeigt.

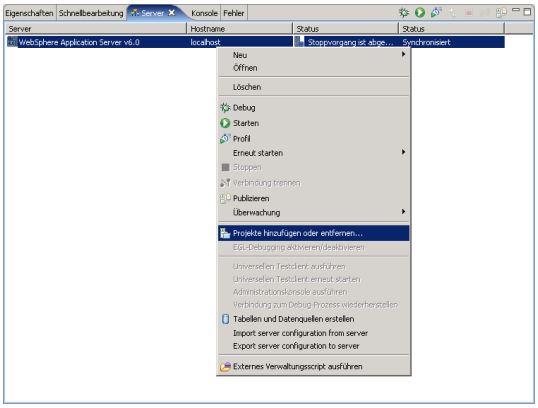

Abbildung 24: Kontextmenü WebSphere Application Server

Wechseln Sie in die Sicht *„Server“*, die sich im Bereich *„Aufgaben“* der Abbildung 1 befindet. Klicken Sie mit der rechten Maustaste auf den Servereintrag *„Websphere Application Server v6.0“* (Abbildung 24) und wählen Sie im Kontextmenü den Eintrag *„Projekte hinzufügen oder entfernen“*.

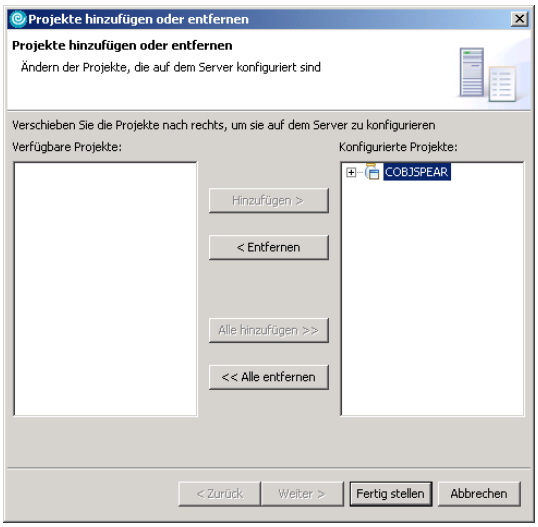

Abbildung 25**:** Projekte hinzufügen oder entfernen

Markieren Sie in dem in Abbildung 25 dargestellten Dialog den Eintrag „COBJSPEAR“ in der Liste *„Verfügbare Projekte“* und wählen Sie anschließend die Taste *„Hinzufügen“*. Der Eintrag erscheint dann in der rechten Liste für *„Konfigurierte Projekte“*. Schließen Sie den Dialog mit der Taste *„Fertig stellen“*. Danach wird der Server automatisch gestartet.
Benutzen Sie das Menü „*Datei* → *Neu* → *Andere*“, um ein neues Objekt zu erstellen.

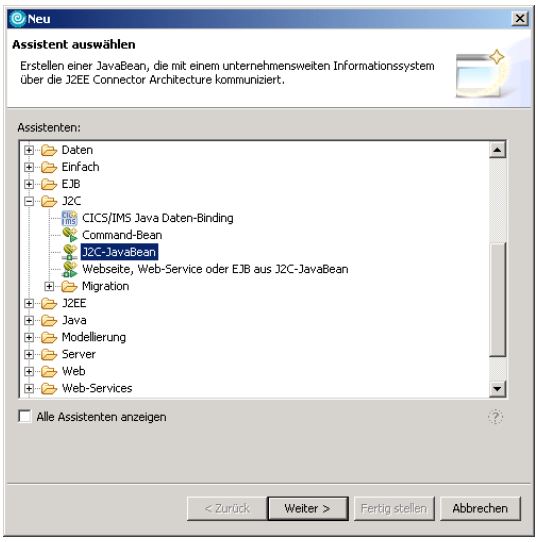

Abbildung 26**:** Neues J2C-JavaBean erstellen

Sie befinden sich jetzt in dem in Abbildung 26 dargestellten Dialog. Erweitern Sie zunächst die Gruppe *„J2C“* durch einen Klick auf das Symbol (+). Wählen Sie danach den Eintrag *„J2C-JavaBean“* und bestätigen Sie die Auswahl mit der Taste *„Weiter“* .

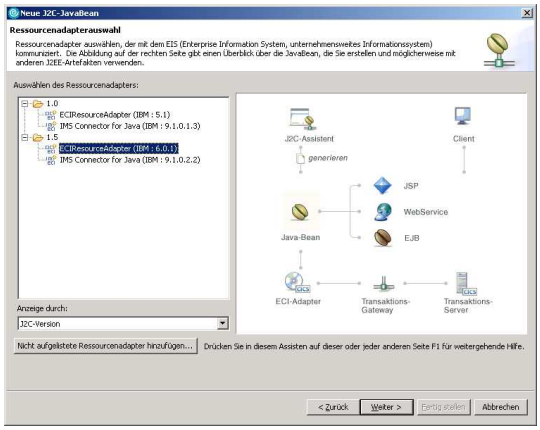

Abbildung 27**:** Ressourceadapter auswählen

Wählen Sie im folgenden Dialog (s. Abbildung 27) den *„ECIRessourceAdapter (IBM:6.0.1)“* in der Gruppe *„1.5“* aus und betätigen Sie die Taste *„Weiter“* .

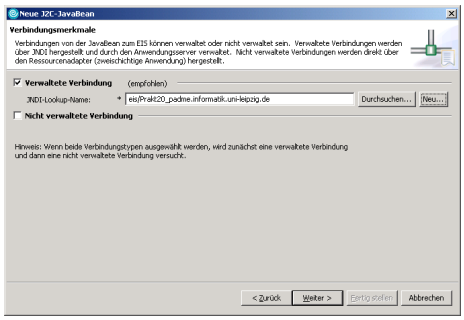

Abbildung 28**:** JNDI-Lookup-Name angeben

Tragen Sie „eis/Prakt20_padme.informatik.uni-leipzig.de“ als *„JNDI-Lookup-Name“* auf dieser Dialogseite ein und prüfen Sie, ob das Feld *„Verwaltete Verbindung“* wie in Abbildung 28 markiert und das Feld *„Nicht verwaltete Verbindung“* nicht markiert ist. Betätigen Sie danach die Taste *„Neu“* rechts neben dem Eingabefeld.

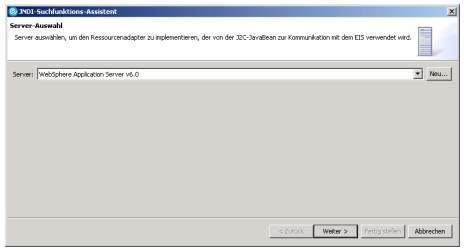

Abbildung 29**:** Applicationserver auswählen

Bestätigen Sie das Dialogfenster *„Server-Auswahl“* mit der Taste *„Weiter“*. Achten Sie darauf, daß der *„WebSphere Application Server v6.0“* ausgewählt ist.

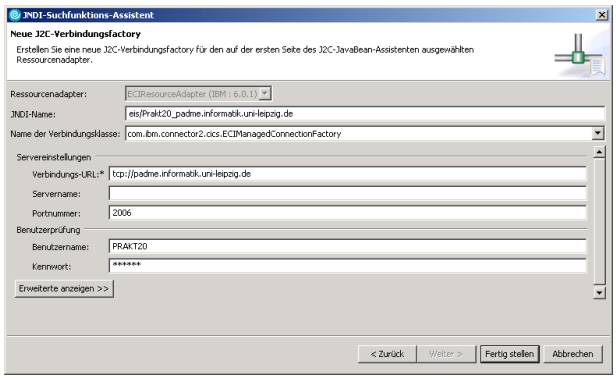

Abbildung 30: Verbindungsparameter für J2C-Verbindungsfactory

Tragen Sie im Dialogfenster *„Neue J2C-Verbindungsfactory“* (s. Abbildung 30) die folgenden Werte ein und schließen Sie das Dialogfenster danach mit der Taste *„Fertig stellen“* .

JNDI-Name	eis/Prakt20_padme.informatik.uni-leipzig.de
Verbindungsklasse	com.ibm.connector2.cics.ECIManagedConnectionFactory
Verbindungs-URL	tcp://padme.informatik.uni-leipzig.de
Portnummer	2006
Benutzername	PRAKT20
Passwort	*Passwort zu ihrem Prak- bzw. Prakt-Account*

Sie befinden Sich jetzt wieder auf der in Abbildung 28 dargestellten Dialogseite. Betätigen Sie die Taste *„Weiter“*.

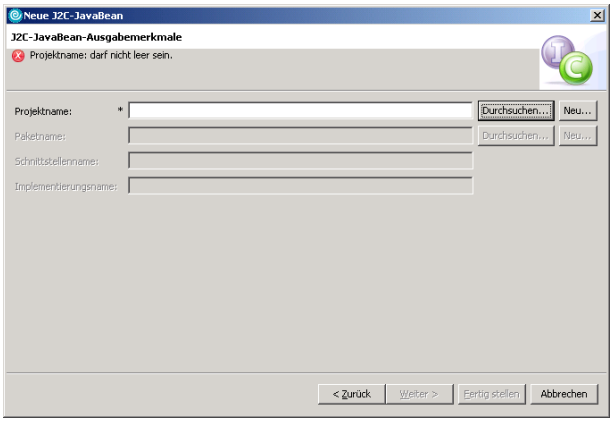

Abbildung 31: Projekt-, Schnittstellen- und Implementierungsname

Die folgende Dialogseite dient der Angabe des Projektes, in dem das J2C-JavaBean erstellt werden soll und der Angabe der Namen und Positionen der zu erstellenden Java-Klassen. Betätigen Sie die Taste *„Durchsuchen“* neben dem Eingabefeld für den Projektnamen.

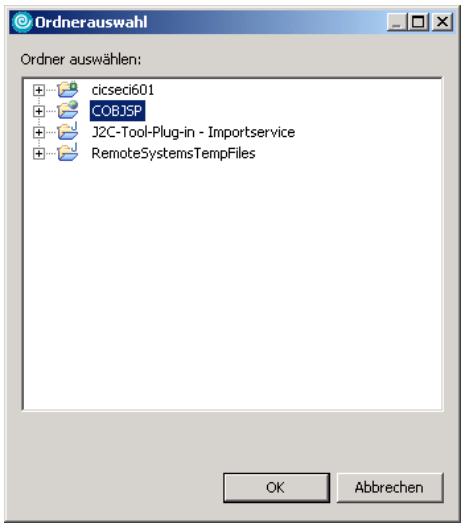

Abbildung 32: Projektauswahl

Wählen Sie im jetzt geöffneten Fenster (Abbildung 32) das Projekt „COBJSP“ aus und bestätigen Sie die Auswahl mit der Taste *„OK“* .
Sie befinden Sich jetzt wieder in dem in Abbildung 31 dargestellten Dialogfenster. Klicken Sie auf die Taste *„Neu“* rechts neben dem Eingabefeld *„Paketname“* .

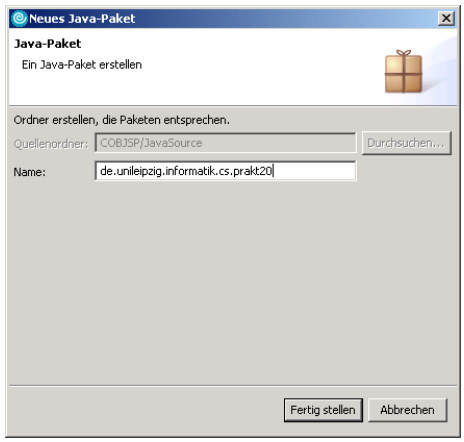

Abbildung 33: Name für neues Java-Paket

Geben Sie im jetzt geöffneten Dialogfenster den Namen für das Paket an, in dem die Schnittstelle und die Implementierungsklasse gespeichert werden sollen. Geben Sie als Paketnamen „de.unileipzig.informatik.cs.prakt20“ an und bestätigen Sie ihre Eingabe mit der Taste *„OK“* .

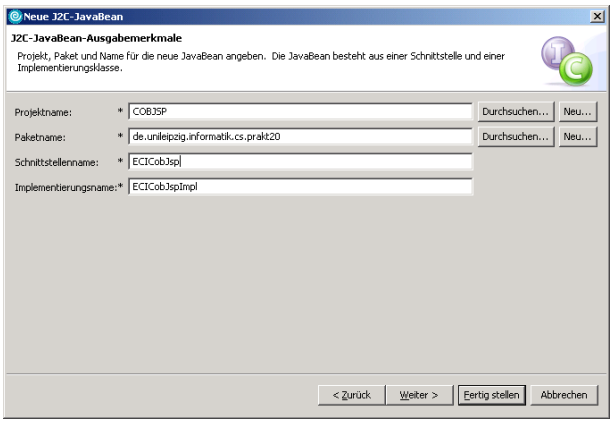

Abbildung 34**:** Name und Speicherort des J2C-JavaBean

Sie befinden Sich erneut in dem in Abbildung 31 dargestellten Dialogfenster. Tragen Sie als Schnittstellenname „ECICobJsp“ ein. Der Implementierungsname wird automatisch als „ECICobJspImpl“ erstellt, so daß die in Abbildung 34 dargestellte Ansicht entsteht. Bestätigen Sie die Eingaben mit der Taste *„Weiter“* .

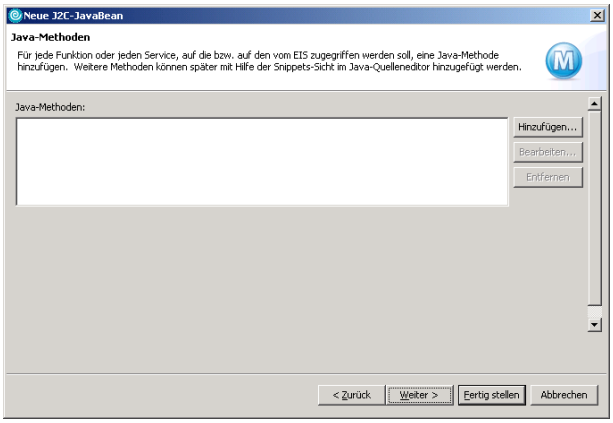

Abbildung 35**:** Java-Methoden des J2C-JavaBean

Das folgende Dialogfenster dient der Definition von Java-Methoden, die das ECI-Bean zur Verfügung stellen soll. Ein solcher Methodenaufruf entspricht später dem Aufruf einer CICS-Transaktion. Klicken Sie mit der Maus auf die Taste *„Hinzufügen“* .

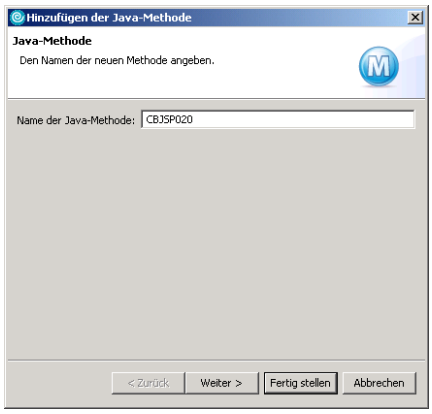

Abbildung 36: Name für neue Java-Methode

Tragen Sie in dem jetzt geöffneten Dialogfenster „CBJSP020“ als Namen für die neu zu erstellende Java-Methode ein und bestätigen Sie die Eingabe mit der Taste „Weiter“ .

Abbildung 37: Neuer Eingabetyp

Der in Abbildung 37 dargestellte Dialog gliedert sich in den oberen Abschnitt *„Eingabetyp“* und den unteren Abschnitt *„Ausgabetyp“*. Wählen Sie hier die Taste *„Neu“* im Abschnitt *„Eingabetyp“*.

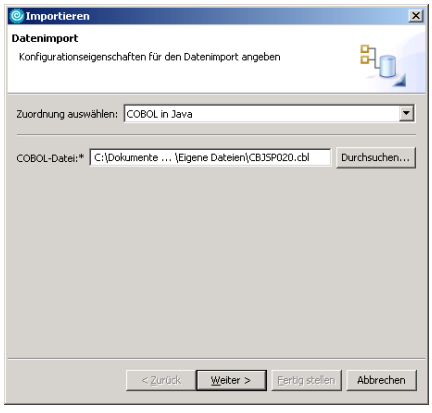

Abbildung 38: Import der COBOL-Anwendung

Betätigen Sie im Dialog *„Dateiimport“* die Taste „*Durchsuchen*“ und wählen Sie Ihre auf der Festplatte gespeicherte Cobol-Datei aus. Bestätigen Sie die Eingabe mit der Taste *„Weiter“* .

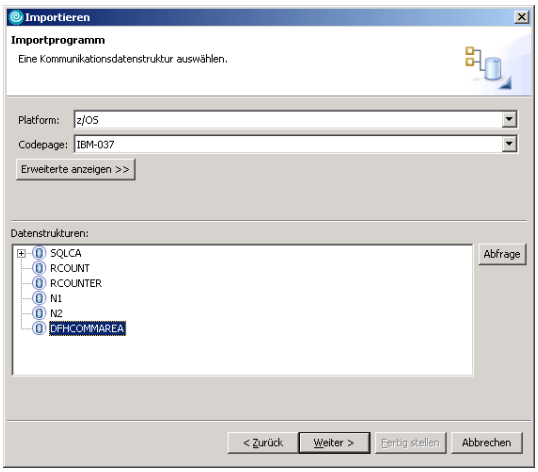

Abbildung 39: Datenstruktur DFHCOMMAREA auswählen

Wählen Sie im Dialog *„Importprgramm“* als Platform „z/OS“ aus. Damit ändert sich die Einstellung für Codepage automatisch auf „IBM-037“. Betätigen Sie danach die Taste *„Abfrage”* am rechten Rand des Dialogfensters. Im links daneben liegenden Feld werden jetzt alle in der COBOL-Datei enthaltenen Datenstrukturen angezeigt. Markieren Sie die Datenstruktur „DFHCOMMAREA“, so daß dieser Eintrag farbig unterlegt ist und vergleichen Sie ihre Eingaben mit dem in Abbildung 39 dargestellten Dialog. Bestätigen Sie danach ihre Eingaben durch Betätigung der Taste *„Weiter“* .

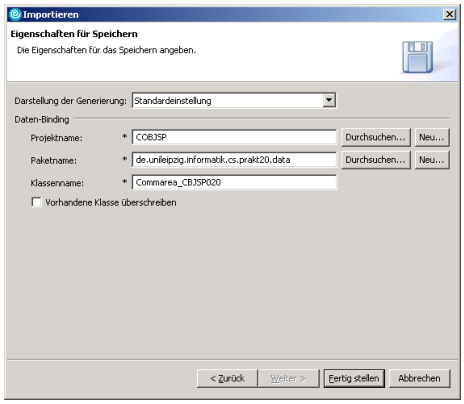

Abbildung 40: Klassenname festlegen

Ändern Sie in dem in Abbildung 40 dargestellten Dialogfenster den Klassennamen auf „Commarea_CBJSP020“ und bestätigen Sie Ihre Eingabe mit der Taste *„Fertig stellen”*.

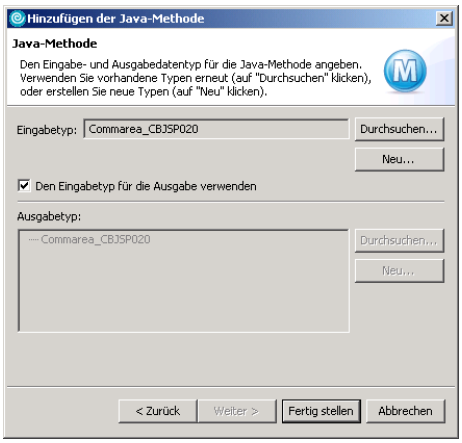

Abbildung 41**:** Ein- und Ausgabetyp festlegen

Sie befinden sich jetzt wieder im Dialog *„Java-Methode“*. Setzen Sie einen Haken in das Feld *„Eingabetyp für die Ausgabe verwenden“*, so daß das Dialogfenster das in Abbildung 41 dargestellte Aussehen erhält. Schließen Sie den Dialog mit der Taste *„Fertig stellen“* .

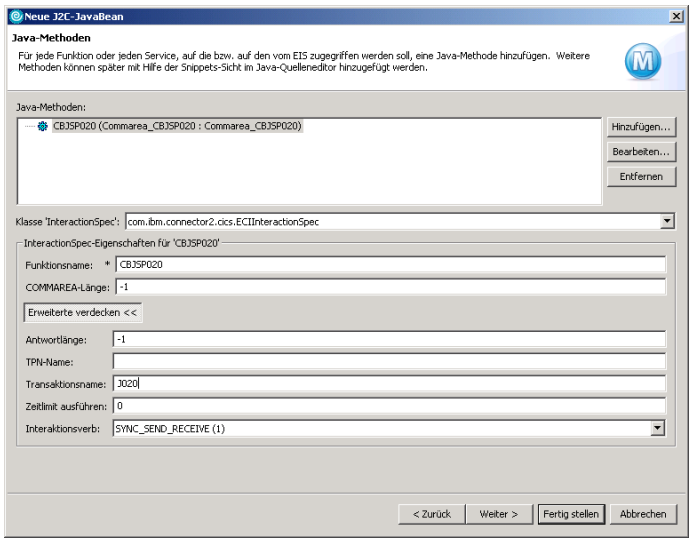

Abbildung 42**:** CICS-Programm und Transaktionsname angeben

Sie befinden sich jetzt wieder im Dialogfenster *„Java-Methoden“* . Tragen Sie als Funktionsname „CBJSP020“ ein und betätigen Sie die Taste *„Erweiterte anzeigen ¿¿“* . Tragen Sie „J020“ als Transaktionsname ein. Mit Betätigung der Taste *„Fertig stellen“* wird das J2C-JavaBean erstellt und die Datei ECICobJspImpl wird geöffnet.
Benutzen Sie das Menü „*Datei* → *Neu* → *Andere*”, um ein neues Objekt zu erstellen.

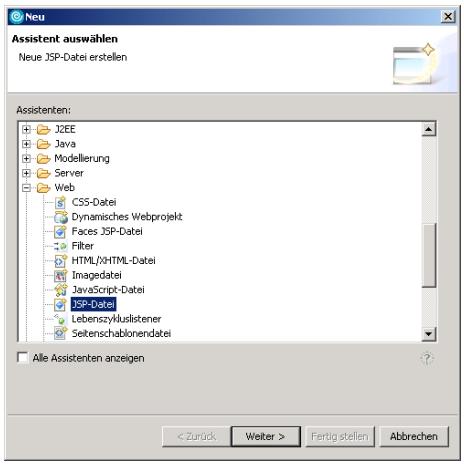

Abbildung 43: neue JSP-Datei

Erweitern Sie die Gruppe *„Web“* durch einen Klick auf das Symbol vor dem Eintrag. Wählen Sie den neu erschienenen Eintrag *„JSP-Datei“* und betätigen Sie die Taste *„Weiter“* (s. Abbildung 43).

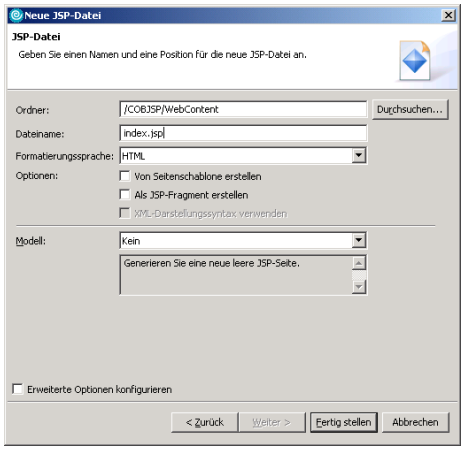

Abbildung 44: Name der neuen JSP-Datei

Tragen Sie im jetzt geöffneten Dialog *„JSP-Datei“* als Dateiname *„index.jsp“* ein und erstellen Sie die JSP-Datei durch Betätigung der Taste *„Fertig stellen“* . Die Datei wird automatisch im Bereich *„Editor“* der Abbildung 1 geöffnet.

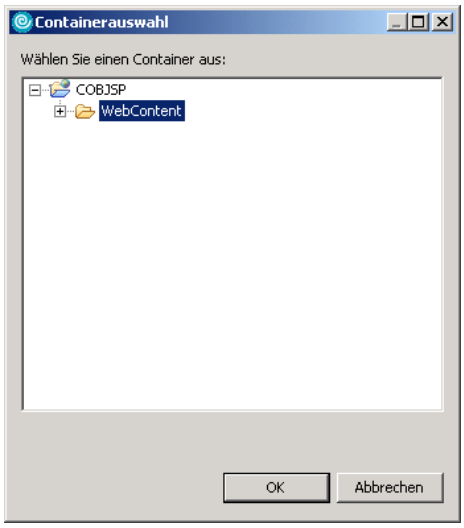

Abbildung 45**:** Container für JSP-Datei auswählen

Betätigen Sie die Taste *„Durchsuchen“,* um in das Dialogfenster in Abbildung 45 zu gelangen. Erweitern Sie die Gruppe COBJSP durch einen Klick auf das vorstehende Symbol. Wählen Sie dann den Eintrag *„WebContent“* aus und schließen Sie den Dialog mit der Taste *„OK“* .

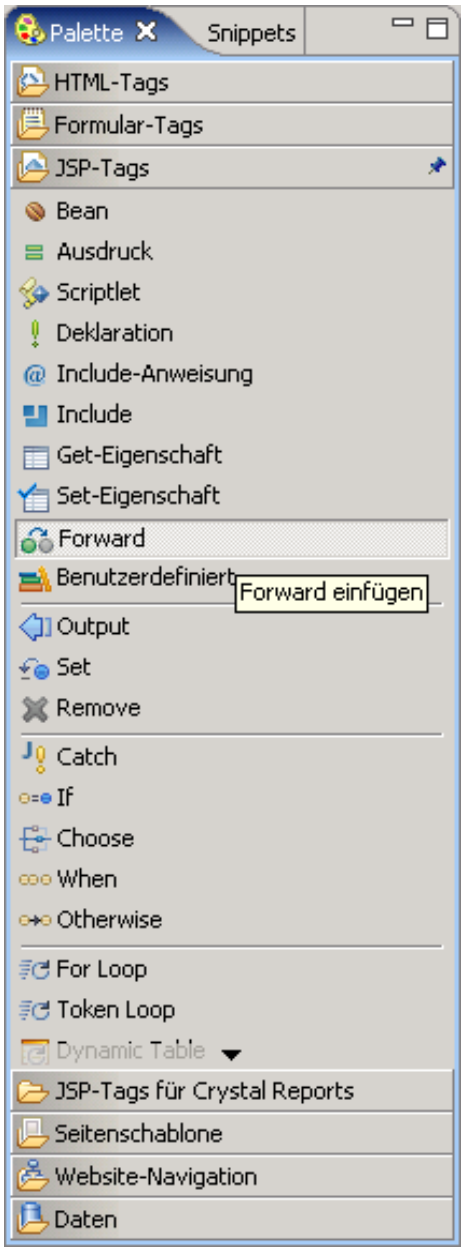

Abbildung 46: Forward einfügen

Wechseln Sie in die Sicht *„Palette“* im Bereich *„Outline“* der Abbildung 1. Erweitern Sie die Gruppe *„JSP-Tags“* durch einfachen Klick und wählen Sie den Eintrag *„Forward“*, so dass dieser wie in Abbildung 46 markiert ist.

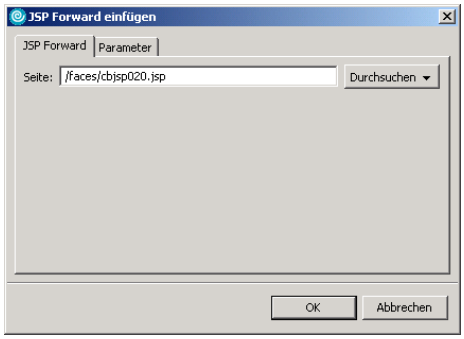

Abbildung 47: Ziel für Forward angeben

Klicken Sie jetzt in die Entwurfsansicht der Datei *„index.jsp“*. Es wird automatisch das

Dialogfenster in Abbildung 47 geöffnet. Tragen Sie „/faces/cbjsp020.jsp“ in das Feld *„Seite“* ein und schließen Sie den Dialog mit der Taste *„OK“*. Speichern Sie ihre Änderungen durch die Eingabe der Tastenkombination *„STRG+S“* oder über den Menüeintrag „*Datei → Speichern*“ .
Benutzen Sie das Menü „*Datei → Neu → Andere*“, um ein neues Objekt zu erstellen.

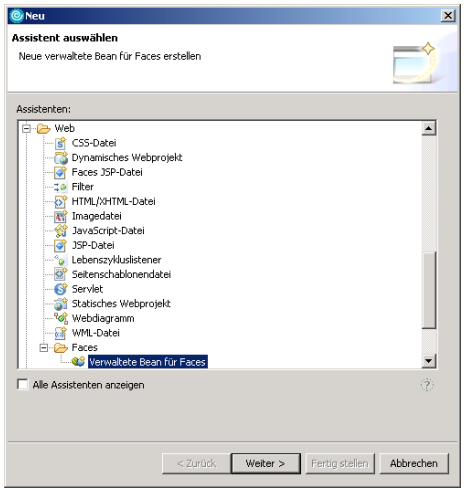

Abbildung 48**:** neues Faces Bean

Erweitern Sie die Gruppe *„Web“* durch einen Klick auf das Symbol vor dem Eintrag. Erweitern Sie danach die Gruppe Faces durch Klick auf das vorstehende Sybol. Wählen Sie den Eintrag *„Verwaltete Bean für Faces“* und betätigen Sie die Taste *„Weiter“* (s. Abbildung 48).

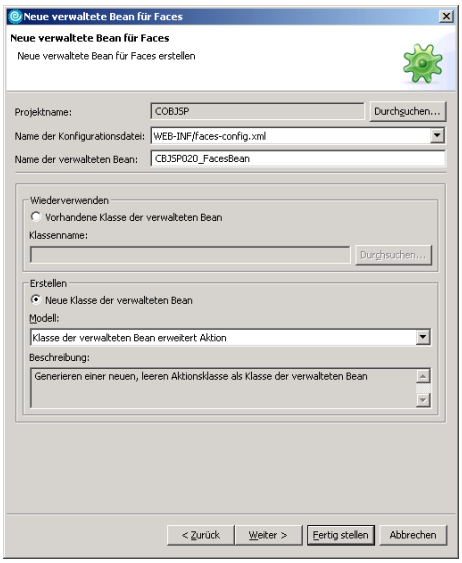

Abbildung 49**:** Name für neues Faces Bean

Der folgende Dialog (s. Abbildung 49) dient der Angabe des Namens des neuen Faces Beans. Tragen Sie „CBJSP020_FacesBean“ in das Feld *„Name der verwaltete Bean“* ein und betätigen Sie die Taste *„Fertig stellen“* .

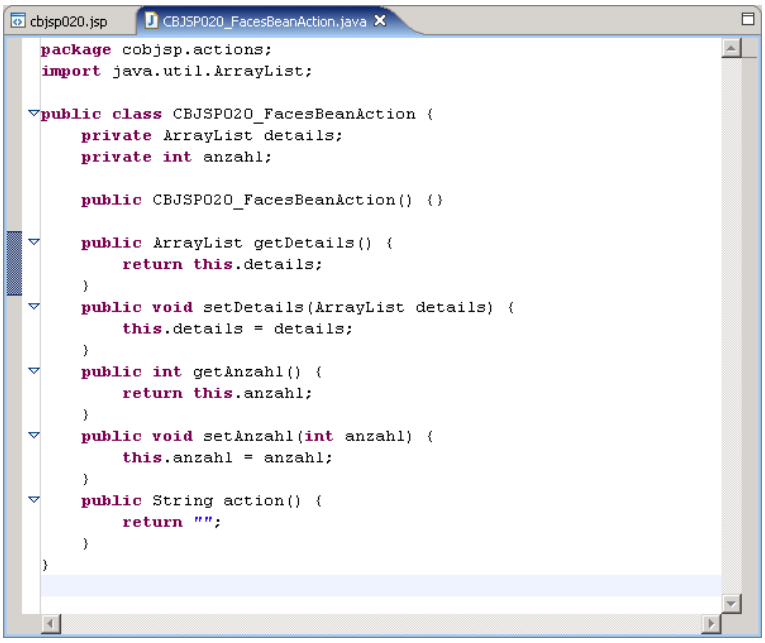

Abbildung 50: Quelltext für Faces Bean

Editieren Sie das jetzt automatisch geöffnete Faces Bean, so dass ihr Bean dem Quelltext in Abbildung 50 entspricht. Speichern Sie ihre Änderungen durch die Eingabe der Tastenkombination *„STRG+S“* oder über den Menüeintrag „*Datei* → *Speichern*“ .
Benutzen Sie das Menü „*Datei* → *Neu* → *Andere*“, um ein neues Objekt zu erstellen.

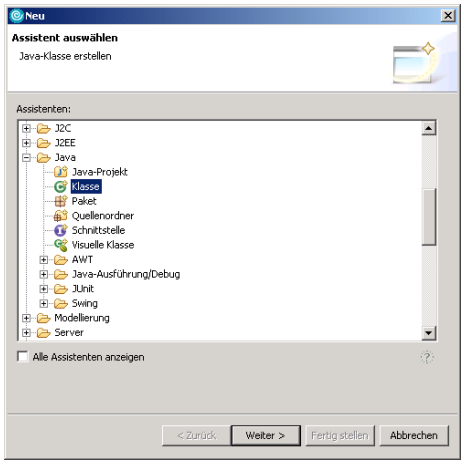

Abbildung 51: neue Javaklasse

Erweitern Sie die Gruppe *„Java“* durch einen Klick auf das Symbol (+) vor dem Eintrag. Wählen Sie den Eintrag *„Klasse“* und betätigen Sie die Taste *„Weiter“* (s. Abbildung 51).

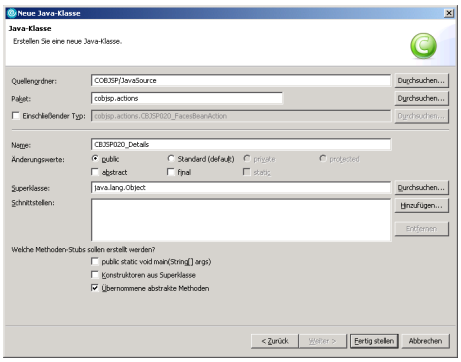

Abbildung 52**:** neue Javaklasse

Tragen Sie in dem in Abbildung 52 dargestellten Dialog *„cobjsp.actions“* in das Feld „Paket“ ein, sowie „CBJSP020_Details“ in das Feld *„Name“* . Betätigen Sie die Taste *„Fertig stellen“* .

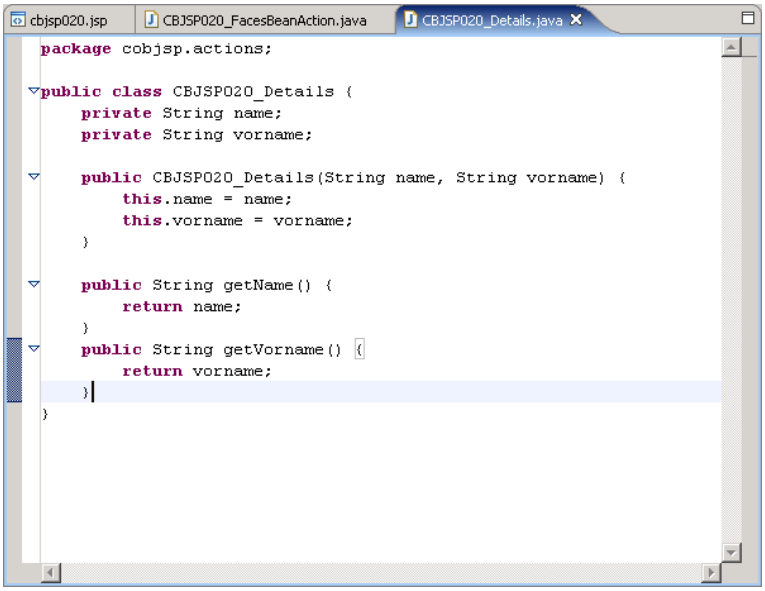

Abbildung53: Quelltext für Detailklasse

Editieren Sie die jetzt automatisch geöffnete Klasse, so dass sie dem Quelltext in Abbildung 53 entspricht. Speichern Sie ihre Änderungen durch die Eingabe der Tastenkombination *„STRG+S”* oder über den Menüeintrag „*Datei → Speichern*”.
Benutzen Sie das Menü ”*Datei → Neu → Andere*”, um ein neues Objekt zu erstellen.

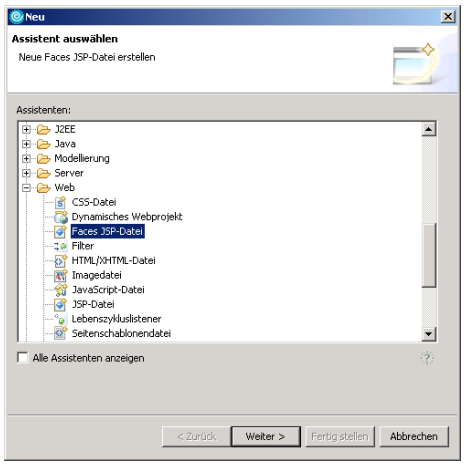

Abbildung 54**:** neue Faces JSP-Datei

Erweitern Sie die Gruppe *„Web“* durch einen Klick auf das Symbol (+) vor dem Eintrag. Wählen Sie den Eintrag Faces JSP-Datei aus und betätigen Sie die Taste *„Weiter“* (s. Abbildung 48).

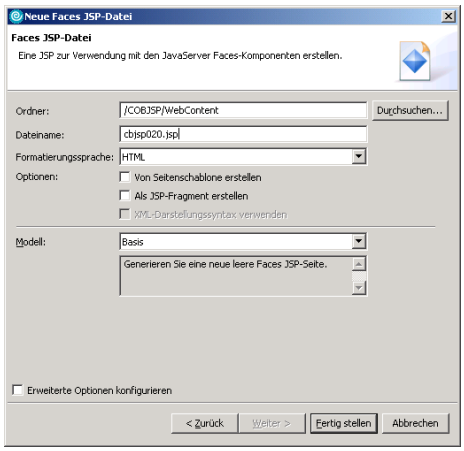

Abbildung 55**:** Name und Speicherort der neuen Faces JSP-Datei

Wählen Sie, wie schon in den Abbildungen 44 und 45 dargestellt, den Container für die neue Faces JSP-Datei aus. Geben Sie als Dateiname *„cbjsp020.jsp“* an und bestätigen Sie ihre Eingaben mit der Taste *„Fertig stellen“* . Die Datei wird danach automatisch im Entwurfsmodus geöffnet.

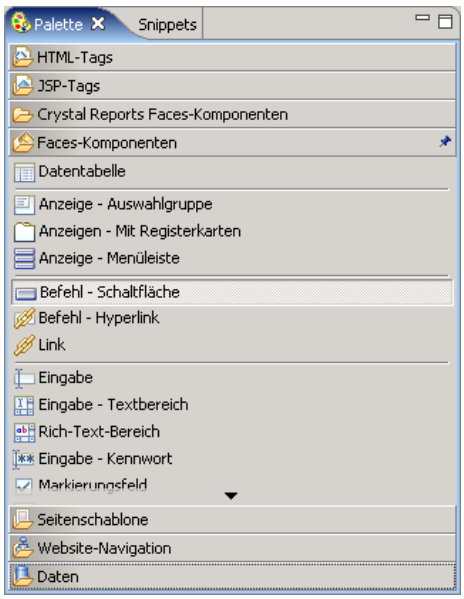

Abbildung 56**:** Faces-Komponenten auswählen

Öffnen Sie, wie in Abbildung 46, die Sicht *„Palette“* und wählen Sie den Eintrag *„Befehl-Schaltfläche“* aus der Gruppe *„Faces-Komponenten“* , so dass dieser Eintrag wie in Abbildung 56 markiert ist.

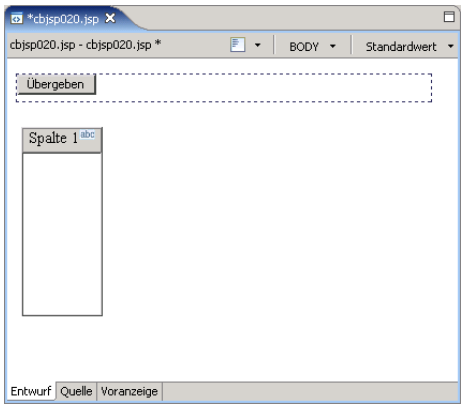

Abbildung 57**:** Layout der Faces JSP-Datei

Klicken Sie danach in die neu erstellte JSP-Datei. Es wird eine Schaltfläche mit der Aufschrift *„Übergeben“* erstellt. Wiederholen Sie diesen Schritt mit der Faces-Komponente *„Datentabelle“*. Danach hat die Faces JSP-Datei das in Abbildung 57 dargestellte Aussehen.

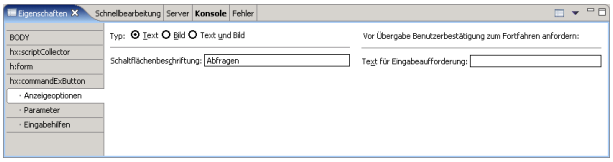

Abbildung 58**:** Schaltflächenbeschriftung

Klicken Sie auf die Schaltfläche *„Übergeben“* im Entwurfsfenster der JSP-Datei. Öffnen Sie jetzt die Sicht *„Eigenschaften“*, die sich im Bereich Aufgaben der Abbildung 1 befindet. In dieser Sicht wird am linken Rand eine Liste der ausgewählten Elemente angezeigt. Wählen Sie den Eintrag *„Anzeigeoptionen“* unterhalb des Eintrags *„hx:commandButton“*. Ändern Sie den Wert für die Schaltflächenbeschriftung auf *„Abfragen“*.

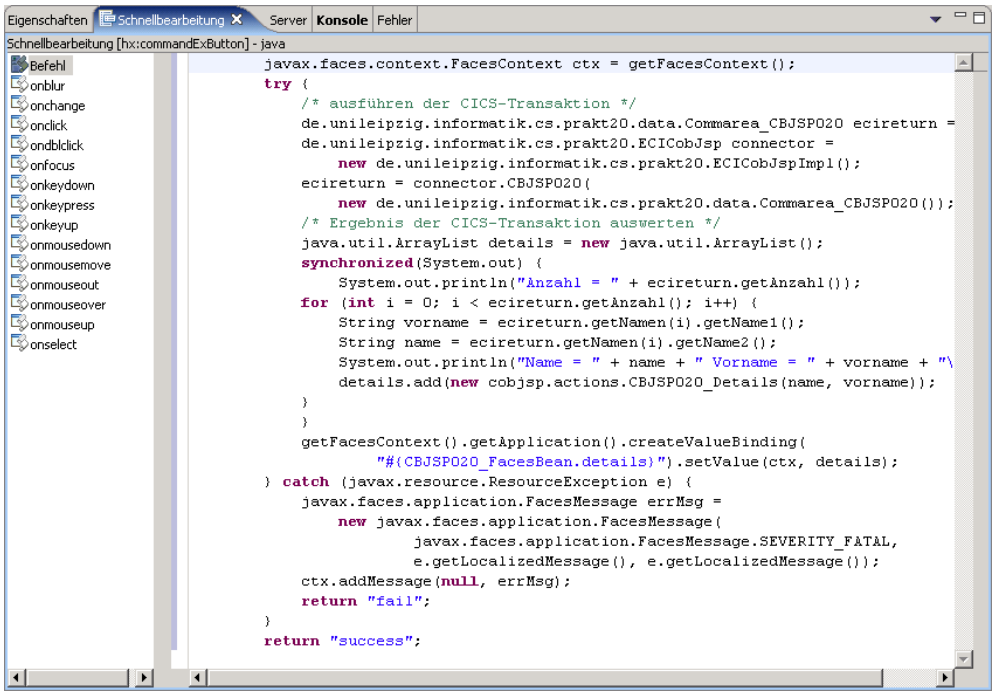

Abbildung 59: Schaltflächenquelltext

Öffnen Sie jetzt die Sicht *„Schnellbearbeitung“* und wählen Sie aus der Liste am linken Rand den Eintrag *„Befehl“*. Fügen Sie den in Abbildung 59 dargestellten Quelltext ein und speichern Sie ihre Änderungen durch die Eingabe der Tastenkombination *„STRG+S“* oder über den Menüeintrag „*Datei → Speichern*“.

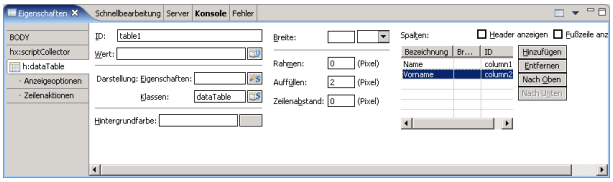

Abbildung 60: Beschriftung der Datentabelle

Wechseln Sie wieder in die Sicht *„Eigenschaften“* und klicken Sie auf die eingefügte Datentabelle mit der Aufschrift *„Spalte 1“*. Wählen Sie in der Liste am linken Rand den Eintrag *„h:dataTable“*. Ändern Sie die Bezeichnung der Spalte in „Name“ durch Klick auf den Wert *„Spalte 1“* in der Tabelle *„Spalten“*. Fügen Sie außerdem eine weitere Spalte mit der Bezeichnung *„Vorname“* hinzu. Vergleichen Sie ihre Ansicht mit Abbildung 60.

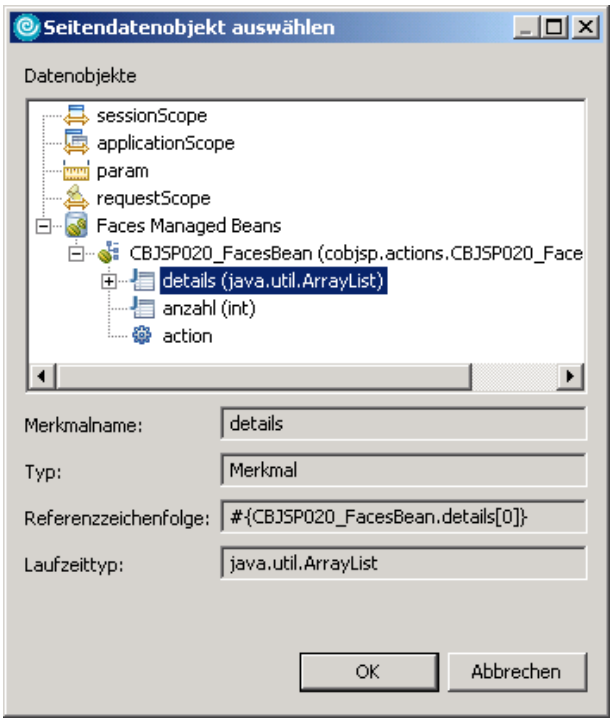

Abbildung 61: Wert für Tabelleninhalt

Klicken Sie jetzt auf das Icon rechts neben dem Feld *„Wert“* in der Sicht *„Eigenschaften“*. Der jetzt geöffnete Dialog (s. Abbildung 61) dient der Angabe des zu verwendenden Feldes des Faces Beans. Erweitern Sie die Gruppen *„Faces Manages Beans“* und *„CBJSP020_FacesBean“* jeweils durch Klick auf das vorstehende Symbol (+) . Markieren Sie, wie in der Abbildung dargestellt, den Eintrag *„details“* und schließen Sie den Dialog durch Betätigen der Taste *„OK“*.

Abbildung 62: Datentyp für Detaildaten

Da der Rückgabedatentyp der Methode *„getDetails()“* des Faces Beans vom Typ *„Arraylist“* ist, muss im jetzt folgenden Dialog (s. Abbildung 62) der Datentyp der in der Liste enthaltenen Elemente angegeben werden. Geben Sie „cobjsp.actions.CBJSP020_Details“ als Datentyp an.

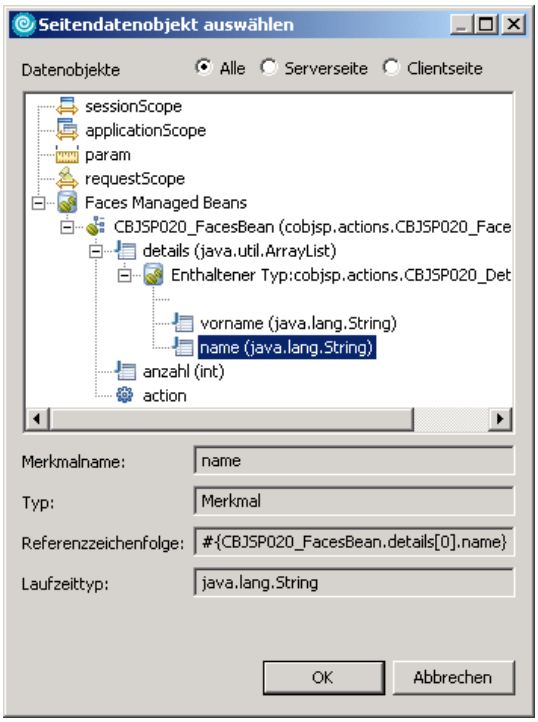

Abbildung 63: Wert für Spalteninhalt

Wechseln Sie in die Sicht *„Palette“* und wählen Sie analog zu Abbildung 56 den Eintrag *„Ausgabe“*. Klicken Sie danach in die Spalte *„Name“* der Datentabelle in der Entwurfsansicht der Datei *„cbjsp020.jsp“*. Wiederholen Sie den Vorgang mit der Spalte *„Vorname“*. Markieren Sie den Eintrag *„outputText“* in der Spalte *„Name“* und wechseln Sie in die Sicht *„Eigenschaften“*. Wählen Sie den Eintrag *„h:outputText“* am linken Rand der Sicht und klicken Sie auf das Icon rechts neben dem Eingabefeld *„Wert“*. Erweitern Sie die Gruppen *„Faces Managed Beans“*, *„CBJSP020_FacesBean“*, *„details“* und *„Enthaltener Typ“* durch Klick auf das vorstehende Symbol (+) . Wählen Sie, wie in Abbildung 63 dargestellt, den Eintrag *„name“* und bestätigen Sie ihre Auswahl mit der Taste *„OK“*. Achten Sie darauf, dass das Eingabefeld *„Wert“* jetzt den Inhalt „#vardetails.name“ besitzt. Wiederholen Sie den Vorhang für die Spalte *„Vorname“*.
Damit ist die Erstellung der Präsentationslogik abgeschlossen. Wechseln Sie in die Sicht *„Server“* und klicken Sie mit der rechten Maustaste auf den Servereintrag *„Websphere Application Server v6.0“*. Wählen Sie im Kontextmenü den Eintrag *„Publizieren“*.

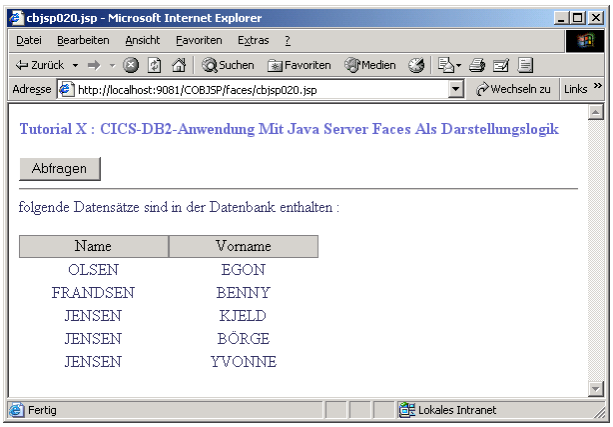

Abbildung 64**:** Ausgabe der Anwendung

Sie können ihre Anwendung dann mit der URL „http://localhost:9080/COBJSP“ aufrufen. Benutzen Sie an Stelle von Port 9080 den Port für HTTP-Transport aus ihrerm WebSphere-Profil.

Aufgabe:	*Erstellen Sie passend zu Ihrem Datenbankschema eine eigene Anwendung.*
Aufgabe:	*Ersetzen Sie in allen Namen und Bezeichnern die Ziffern 020 durch die Nummer Ihres Praktxx bzw. Prakxxx-Account.*
Aufgabe:	*Erstellen Sie einen Screenshot der Ausgabe in Ihrem Browserfenster. Die Ausgabe muss Ihren Namen enthalten, wenn die Aufgabe von mehreren Personen bearbeitet wird müssen alle Namen enthalten sein.* *Benutzen Sie das kompakte JPEG-Format zur Erstellung des Screenshot, die Datei sollte eine Größe von 90 KBytes nicht überschreiten.*