

eTutor

Benutzerhandbuch

Benutzerhandbuch JDBC

Georg Nitsche

Version 1.0
Stand März 2006

INSTITUT FÜR WIRTSCHAFTSINFORMATIK

The logo for 'dke' consists of the lowercase letters 'dke' in a bold, green, sans-serif font. The letters are positioned on a light gray rectangular background that is wider than the text itself.
Data & Knowledge Engineering

Versionsverlauf:

<i>Version</i>	<i>Autor</i>	<i>Datum</i>	<i>Änderungen</i>
1.0	gn	06.03.2006	Fertigstellung der ersten Version

Inhaltsverzeichnis:

1.	Einleitung.....	1
2.	Systemnutzung.....	2
2.1.	Studentensicht.....	2
2.2.	Assistentensicht	7

Abbildungsverzeichnis:

Abbildung 2.1:	Eingabe einer Studentenlösung.....	3
Abbildung 2.2:	Feedback zu korrekter Lösung (keine Diagnose)	4
Abbildung 2.3:	Feedback zu korrekter Lösung (Abgabe).....	4
Abbildung 2.4:	Feedback zu fehlerhafter Lösung (keine Diagnose)	5
Abbildung 2.5:	Feedback zu fehlerhafter Lösung (niedrige Diagnosestufe)	5
Abbildung 2.6:	Feedback zu fehlerhafter Lösung (Abgabe).....	6
Abbildung 2.7:	Feedback zu nicht ausführbarer Lösung	6
Abbildung 2.8:	Feedback zu fehlerhafter Lösung (mittlere Diagnosestufe).....	7

1. Einleitung

Dieses Dokument beinhaltet eine Beschreibung der Benutzerschnittstelle des JDBC-Moduls, das als Teilbereich des eTutor-Systems die Analyse und Bewertung von JDBC-Übungsaufgaben erlaubt. Studenten können über diese Schnittstelle eine Datei, die Code in der Programmiersprache Java enthält, an das eTutor-System übermitteln und ausführen lassen. Je nach Aufgabenstellung müssen Studenten durch das ausgeführte Programm bestimmte Manipulationen an einem vordefinierten Datenbankschema vornehmen. JDBC (Java Database Connectivity) stellt hierbei die Schnittstelle dar, die den Zugriff und Operationen auf dem Datenbankschema ermöglicht.

Nachdem das JDBC-Modul als Teilbereich in das eTutor-System integriert ist, sein für eine grundsätzliche Beschreibung der Einbettung in das Gesamtsystem hiermit auf das allgemeine Benutzerhandbuch des eTutor-Systems verwiesen. Dies betrifft vor allem das Aufrufen von Aufgaben durch Studenten. Kapitel 1 beschäftigt sich in diesem Zusammenhang mit der Benutzeroberfläche, die einem Studenten für die Ausarbeitung einer Aufgabe zum Aufgabengebiet JDBC zur Verfügung steht

2. Systemnutzung

Dieses Kapitel dient einem Überblick über die Möglichkeiten, die Benutzern des eTutor-Systems für die Bearbeitung von Übungsbeispielen aus dem Aufgabengebiet JDBC zur Verfügung stehen. Abschnitt 2.1 enthält Informationen zur Ausarbeitung von Aufgaben, die eine Zuteilung von Übungsbeispielen zu Studenten darstellen. Im Gegensatz zur Ausarbeitung von Übungsbeispielen durch Studenten stellt das JDBC-Modul derzeit noch keine Benutzerschnittstelle für die Eingabe von Übungsbeispielen durch Assistenten zur Verfügung (siehe Abschnitt 2.1).

2.1. Studentensicht

Abbildung 2.1 zeigt die Benutzerschnittstelle für die Ausführung einer Datei zu einem Übungsbeispiel. Die Datei wird hochgeladen und muss eine Java-Klasse enthalten, die das vom JDBC-Modul vorgegebene Java-Interface *JDBCExecutor* und die darin enthaltene Methode *execute* implementieren muss:

```
package etutor.modules.jdbc.analysis;

import java.io.PrintStream;
import java.sql.Connection;

public interface JDBCExecutor {

    public void execute(Connection con, PrintStream out) throws Exception;

}
```

Die im folgenden Programmtext definierte Klasse *JDBCProgram* stellt ein Beispiel für eine Implementierung von diesem Interface dar.

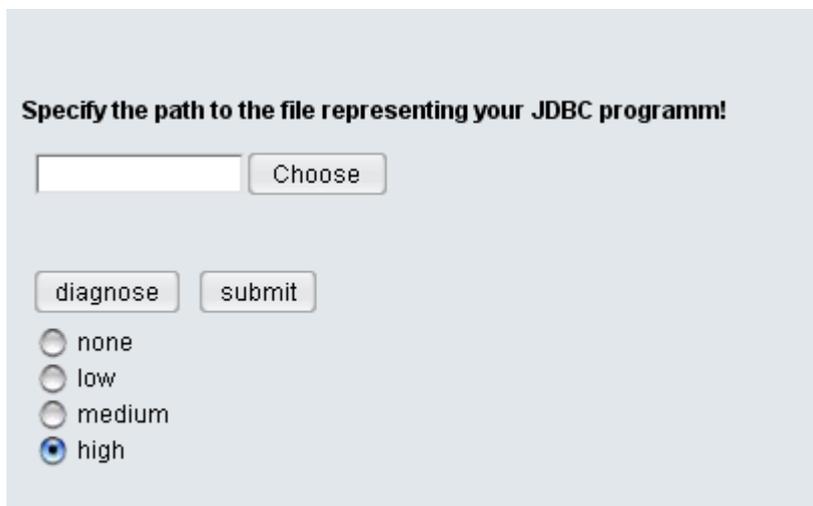
```
import java.io.PrintStream;
import java.sql.Connection;
import etutor.modules.jdbc.analysis.JDBCExecutor;

public class JDBCProgram implements JDBCExecutor {

    public void execute(Connection conn, PrintStream out) throws Exception {
        //perform database operations here
    }
}
```

```
}  
}
```

Die Aufgabenstellung des jeweiligen Übungsbeispiels wird in einem eigenen Fenster dargestellt, dessen Anzeige im allgemeinen Benutzerhandbuch des eTutor-Systems beschrieben wird. Der Aufgabenstellung entsprechend müssen in der implementierten Methode *execute* die geforderten Manipulationen im vorgegebenen Datenbankschema vorgenommen werden. Zu verwenden ist dabei das in der Methode übergebene *Connection*-Objekt. Das *PrintStream*-Objekt kann verwendet werden, um Informationen, etwa zur Protokollierung von Programmzuständen, auszugeben. Diese Informationen werden bei einer Programmausführung als Bestandteil des Feedbacks angezeigt.



Specify the path to the file representing your JDBC program!

Choose

diagnose submit

none
 low
 medium
 high

Abbildung 2.1: Eingabe einer Studentenlösung

In Abbildung 2.1 zu sehen sind außerdem die Möglichkeiten, die das JDBC-Modul für die Ausführung eines JDBC-Programms zur Verfügung stellt:

- *Diagnose (Diagnose)*: In diesem Modus können JDBC-Programme auf vier Diagnosestufen getestet werden. Je höher die Diagnosestufe ist, desto mehr Informationen zu eventuellen Fehlerquellen sind im Feedback enthalten. Fehler ergeben sich aus der Abweichung des tatsächlichen Datenbankschemas nach der Ausführung und dem geforderten Datenbankschema, das sich aus den Forderungen der Aufgabenstellung ergibt.
- *Abgabe (Submit)*: Eine abgegebene Lösung wird im eTutor-System gespeichert und bewertet.

Diese Ausführungsmöglichkeiten stellt das JDBC-Programm grundsätzlich zur Verfügung, die bei der Ausarbeitung einer Aufgabe tatsächlich zur Verfügung stehenden Ausführungsmöglichkeiten hängen allerdings von Faktoren ab, die im allgemeinen Benutzerhandbuch des eTutor-Systems nachzulesen sind.

Abbildung 2.2 demonstriert die Abgabe einer korrekten Lösung auf niedrigster Diagnosestufe. Das dabei angezeigte Feedback enthält lediglich den Hinweis darauf, dass das JDBC-Programm offensichtlich korrekt ist, da das Datenbankschema nach der Ausführung der Methode *execute* die geforderten Informationen enthält. Davon abgesehen werden, wie bereits erwähnt, jegliche Informationen ausgegeben, die in das *PrintStream*-Objekt geschrieben wurden. Im Gegensatz dazu wird bei einer Abgabe die erreichte Punkteanzahl angezeigt (siehe Abbildung 2.3).

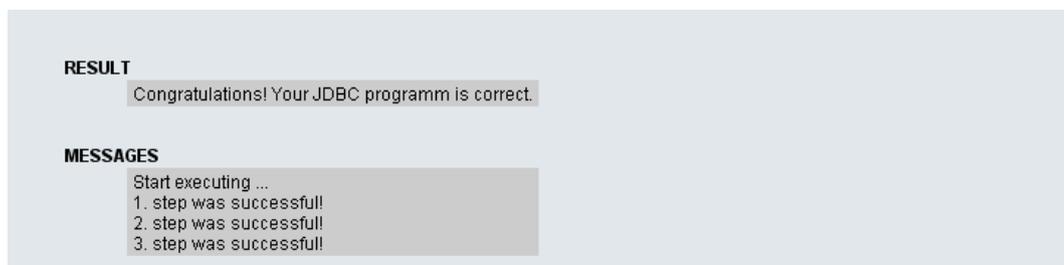


Abbildung 2.2: Feedback zu korrekter Lösung (keine Diagnose)

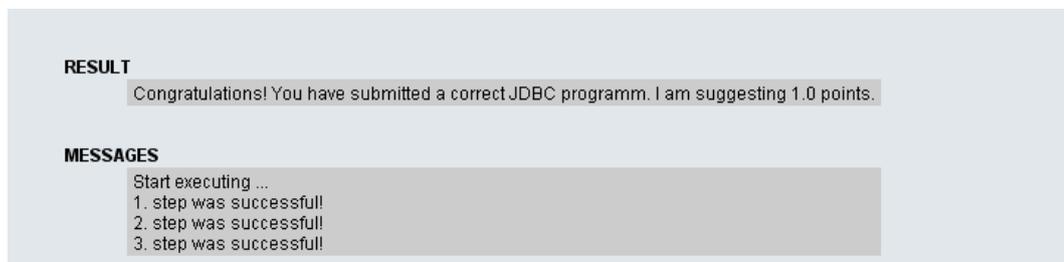


Abbildung 2.3: Feedback zu korrekter Lösung (Abgabe)

Nachfolgend werden Beispiele für Feedback auf fehlerhafte Lösungen gezeigt. Im ersten Beispiel wird angenommen, dass das Programm zwar ausführbar ist, aber in einem JDBC-Konstrukt eine ungültige SQL-Anweisung enthält. In diesem Fall erhält der Student auf niedrigster Diagnosestufe ein Feedback, das dem in Abbildung 2.4 dargestellten Feedback entspricht. Zu sehen ist, dass die gelieferten Informationen über die Fehlerursache reduziert gehalten sind. Wird das Programm jedoch auf einer höheren Diagnosestufe ausgeführt, so wird der Grad an Information dementsprechend erhöht (siehe Abbildung 2.5). Bei einer Abgabe

enthält das Feedback die zusätzliche Information, dass für die Lösung keine Punkte vergeben wurden (siehe Abbildung 2.6).

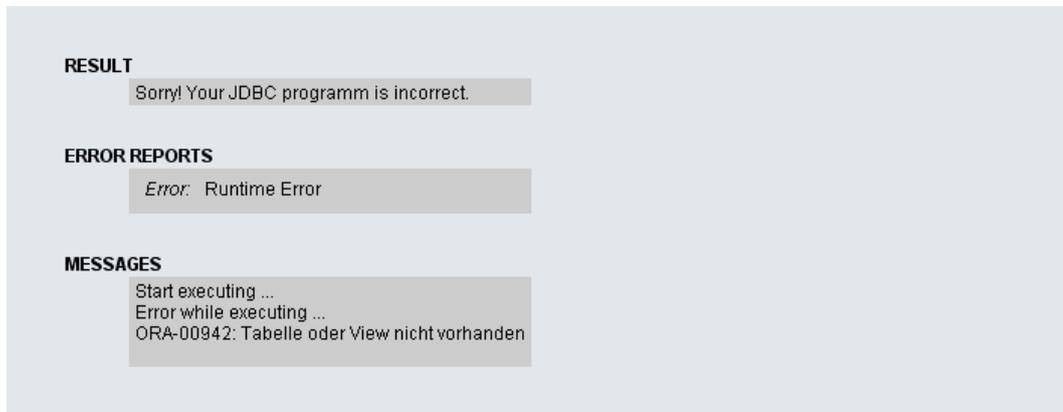


Abbildung 2.4: Feedback zu fehlerhafter Lösung (keine Diagnose)

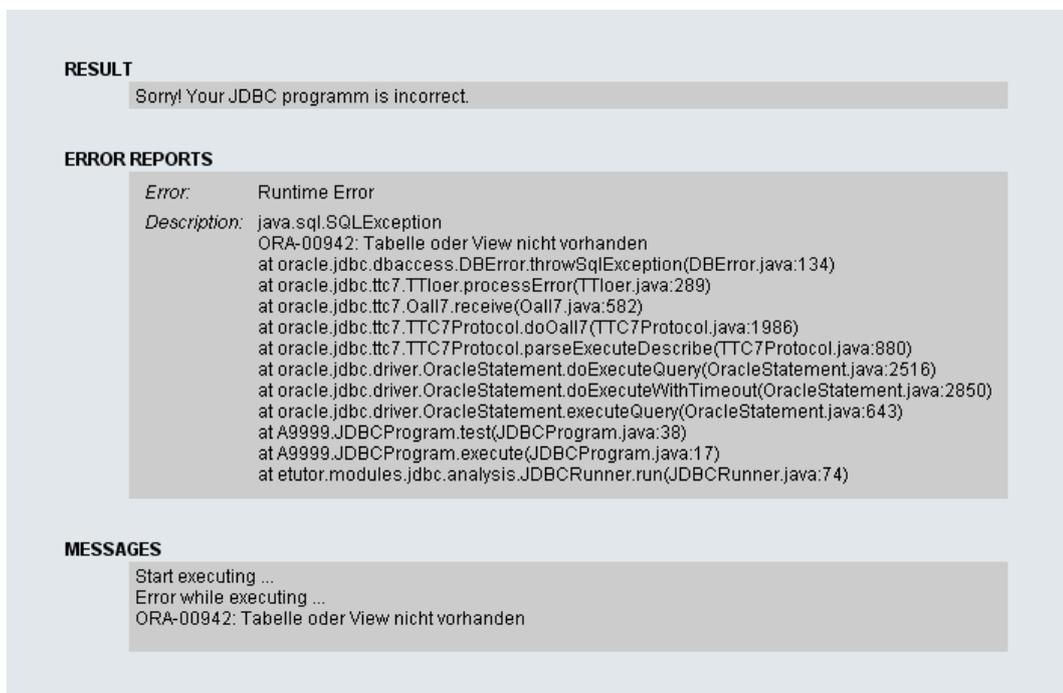


Abbildung 2.5: Feedback zu fehlerhafter Lösung (niedrige Diagnosestufe)

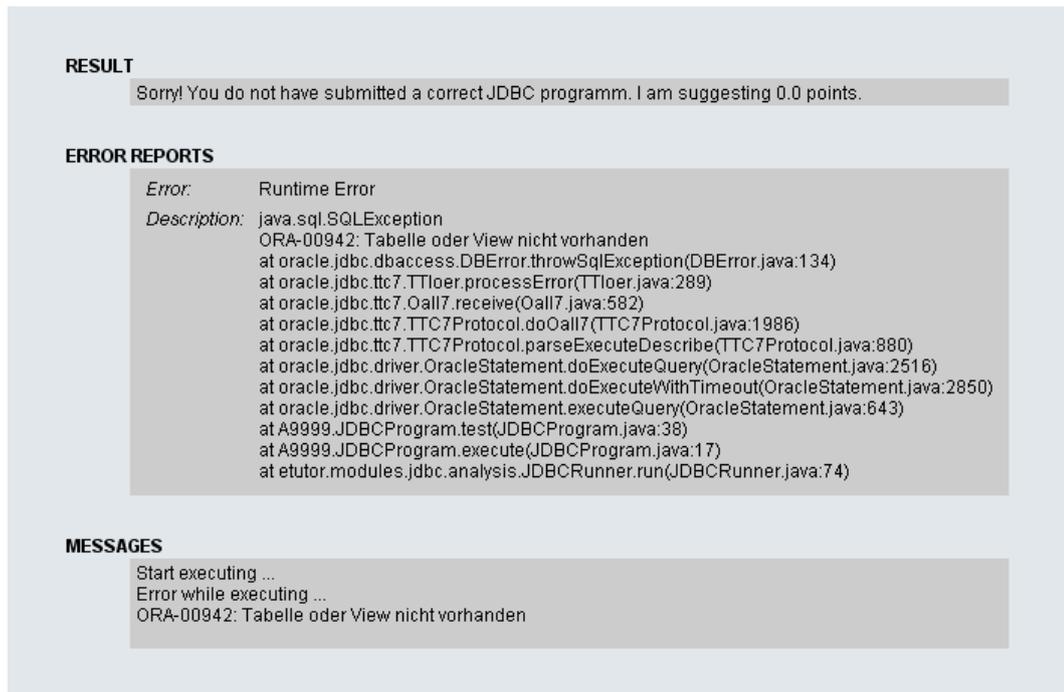


Abbildung 2.6: Feedback zu fehlerhafter Lösung (Abgabe)

Abschließend seien noch beispielhaft zwei spezielle Fehlerfälle demonstriert. Abbildung 2.7 zeigt das Feedback, das angezeigt wird, wenn die Übersetzung des Programms in Maschinencode fehlschlägt. In diesem Fall wurde eine Datei hochgeladen, in der eine Klasse *JDBCExecutor2* definiert wurde. Der Java-Compiler, der vom JDBC-Modul verwendet wird, erwartet in diesem Fall eine Datei mit der Bezeichnung *JDBCExecutor2.java*, tatsächlich wurde aber eine Datei mit der Bezeichnung *JDBCExecutor.java* hochgeladen.

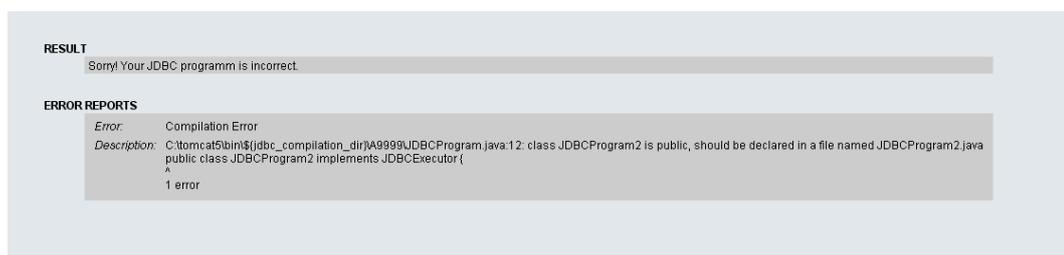


Abbildung 2.7: Feedback zu nicht ausführbarer Lösung

Abbildung 2.8 zeigt ein Beispiel für die Verletzung von Sicherheitseinschränkungen durch das ausgeführte JDBC-Programm. Zu den Sicherheitseinschränkungen, die vom JDBC-Modul vorgegeben werden, zählen Operationen wie etwa der Zugriff auf das Dateisystem des JDBC-Moduls.

```
RESULT
  Sorry! Your JDBC programm is incorrect.

ERROR REPORTS
  Error: Runtime Error
  Description: java.lang.SecurityException
               Requested permission (java.io.FilePermission C:\tomcat5\bin\dir read) not granted!at
               etutor.modules.jdbc.analysis.JDBCSecurityManager.checkPermission(JDBCSecurityManager.java:48)
               at java.lang.SecurityManager.checkRead(SecurityManager.java:863)
               at java.io.File.exists(File.java:678)
               at java.io.Win32FileSystem.canonicalize(Win32FileSystem.java:358)
               at java.io.File.getCanonicalPath(File.java:513)
               at java.io.FilePermission$1.run(FilePermission.java:218)
               at java.security.AccessController.doPrivileged(Native Method)
               at java.io.FilePermission.init(FilePermission.java:212)
               at java.io.FilePermission.(FilePermission.java:264)
               at java.lang.SecurityManager.checkWrite(SecurityManager.java:954)
               at java.io.File.mkdir(File.java:1097)
               at A9999.JDBCProgram.execute(JDBCProgram.java:14)
               at etutor.modules.jdbc.analysis.JDBCRunner.run(JDBCRunner.java:74)
```

Abbildung 2.8: Feedback zu fehlerhafter Lösung (mittlere Diagnosestufe)

2.2. Assistentensicht

In der Assistentensicht des eTutor-Systems sind grundsätzlich Möglichkeiten für die Spezifikation von Übungsbeispielen vorgesehen. Die Spezifikation von JDBC-Übungsbeispielen wird allerdings derzeit vom JDBC-Modul noch nicht unterstützt.