**实验三 jdbc驱动和插入操作**

## 一、实训目标

1．运用JDBC驱动方式连接数据库

2. 访问数据库，执行数据插入操作

## 二、实训内容

**1.**打开mysql（Navicat），建立数据库scms ，建3个表。

（1）课程表course，表结构如图1所示。各个字段的含义是：课程号、课程名、开课专业、学时、学分、任课教师编号、课程介绍。其中课程号cno是主键，任课教师编号tno是外键。样本数据如图2所示。

（2）用户登录表login，表结构如图3所示。各个字段的含义是：用户名、密码。其中用户名username是主键。样本数据如图4所示。

（3）教师表teacher，表结构如图5所示。各个字段的含义是：任课教师编号、姓名、密码、性别、职称、所在部门、电话、邮箱。其中任课教师编号tno是主键。样本数据如图6所示。

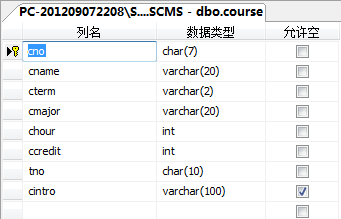


图1 course表的结构

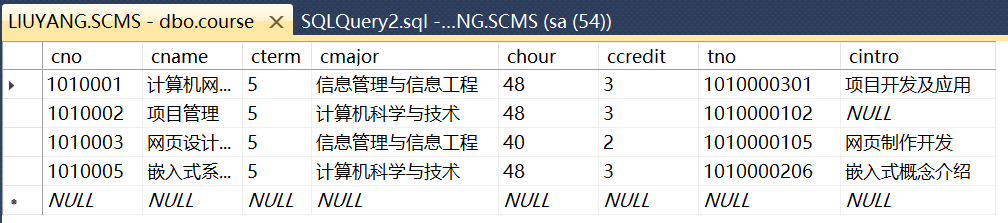


图2 course表样本数据

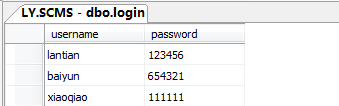
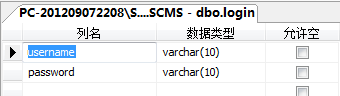


图3 login表的结构 图4 login表样本数据

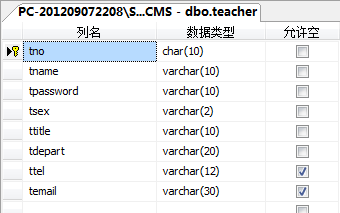


图5 teacher表的结构

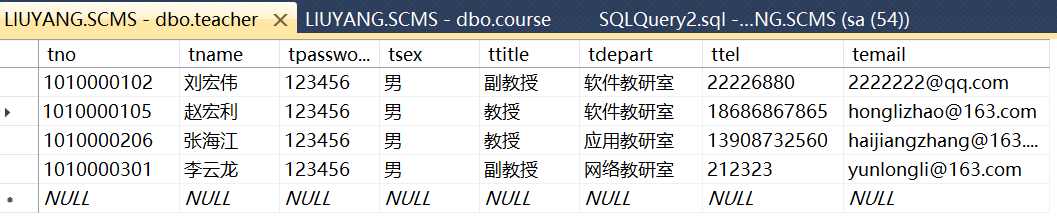


图6 teacher表样本数据

|  |
| --- |
| 此处插入sql代码——建库、建表、插入记录。  CREATE TABLE course (  cno char(7) PRIMARY KEY,  cname varchar(20),  cterm int,  cmajor varchar(20),  chour int,  ccredit int,  tno char(10) NOT NULL,  cintro varchar(100)  );  CREATE TABLE login (  username VARCHAR(10) PRIMARY KEY,  password VARCHAR(10) NOT NULL  );  CREATE TABLE teacher (  tno CHAR(10) PRIMARY KEY,  tname VARCHAR(10) NOT NULL,  tpassword VARCHAR(10) NOT NULL,  tsex VARCHAR(2),  ttitle VARCHAR(10),  tdepart VARCHAR(20),  ttel VARCHAR(12),  temail VARCHAR(30)  );  SELECT\*from teacher  INSERT INTO course (cno, cname, cterm, cmajor, chour, ccredit, tno, cintro)  VALUES  ('1010001', '计算机网络', 5, '信息管理与信息工程', 48, 3, '1010000301', '项目开发及应用'),  ('1010002', '项目管理', 5, '计算机科学与技术', 48, 3, '1010000102', NULL),  ('1010003', '网页设计', 5, '信息管理与信息工程', 40, 2, '1010000105', '网页制作开发'),  ('1010005', '嵌入式系统', 5, '计算机科学与技术', 48, 3, '1010000206', '嵌入式概念介绍');  INSERT INTO login (username, password)  VALUES ('lantian', '123456'), ('baiyun', '654321'), ('xiaoxiao', '111111');  INSERT INTO teacher (tno, tname, tpassword, tsex, ttitle, tdepart, ttel, temail)  VALUES  ('1010000102', '刘宏伟', '123456', '男', '副教授', '软件教研室', '22226880', 'honglizhao@163.com'),  ('1010000105', '赵宏利', '123456', '男', '教授', '软件教研室', '18686878685', 'honglizhao@163.com'),  ('1010000206', '张海江', '123456', '男', '教授', '软件教研室', '13908732865', 'hajiangzhang@163.com'),  ('1010000301', '李云龙', '123456', '男', '副教授', '网络教研室', '13908732560', 'yunglongli@163.com'); |

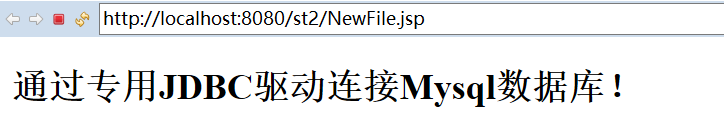
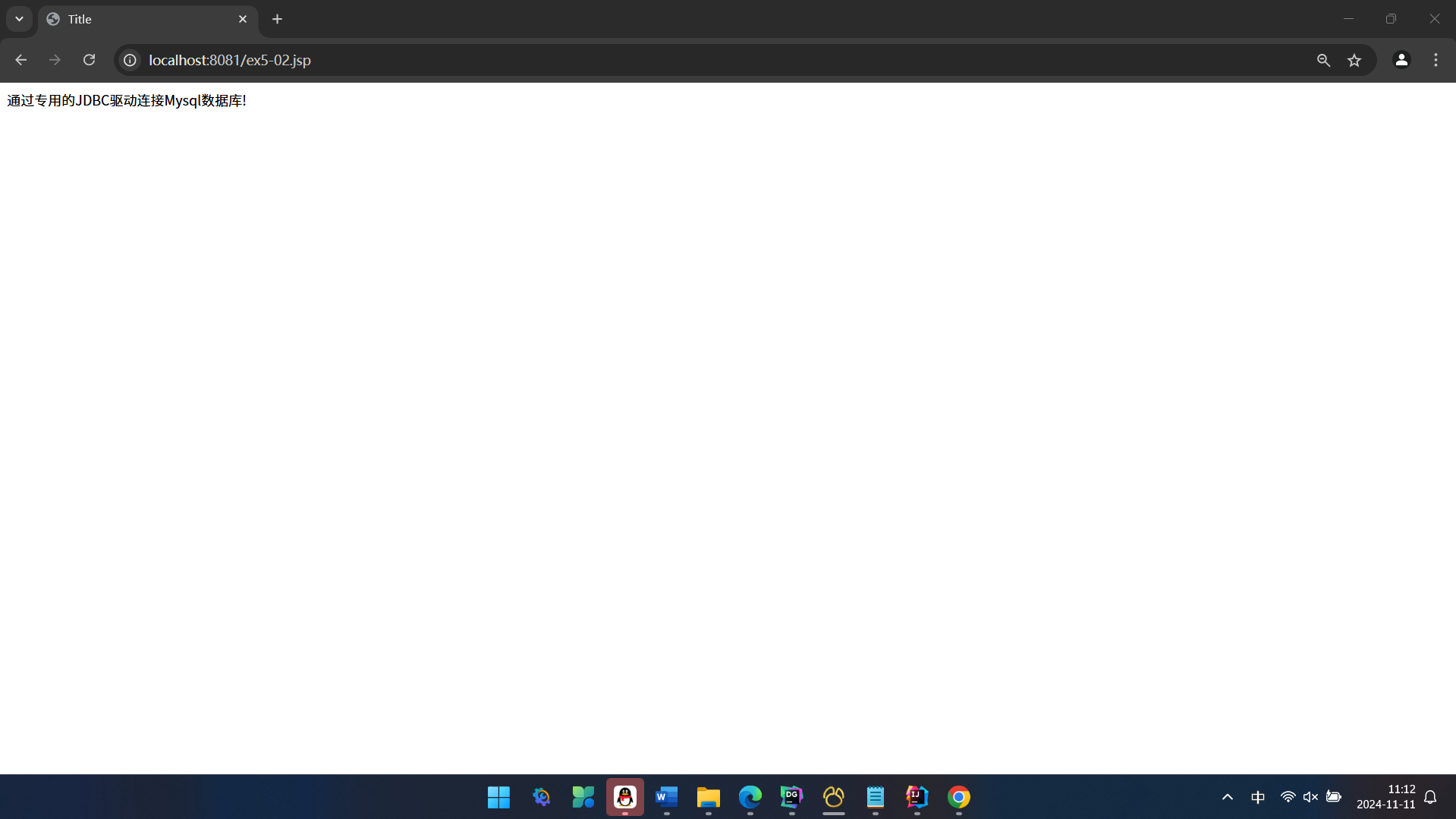
2. 编码创建文件ex5-02.jsp,运用JDBC驱动方式连接数据库。运行ex5-02.jsp文件，如果一切正常，就会看到如图7所示的连接成功页面，否则会看到抛出异常的页面。比如出现加载驱动异常、TCP/IP通信连接异常等等。

图7 运行页面效果



|  |
| --- |
| 此处插入代码和运行页面截图。  package dao;   import java.sql.\*;  */\*\*  \* 基础数据访问对象（DAO），提供数据库连接、执行SQL语句和关闭资源的方法。  \*/* public class BaseDao {   // 初始化连接对象，用于与数据库建立连接  private Connection connection = null;   // 预编译的 SQL 语句对象，可以用来执行参数化的 SQL 查询或更新  private PreparedStatement preparedStatement = null;   // 结果集对象，存储从数据库查询返回的数据  private ResultSet resultSet = null;   */\*\*  \* 加载数据库驱动并建立连接。  \*  \** ***@return*** *如果成功建立连接，则返回 true；否则返回 false。  \*/* public boolean getConnection() {  try {  // 注册 JDBC 驱动  Class.*forName*("com.mysql.cj.jdbc.Driver");   // 建立数据库连接  connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/scms", "root", "123456");   return true;  } catch (SQLException e) {  // 处理 SQL 异常  e.printStackTrace();  return false;  } catch (ClassNotFoundException e) {  // 处理找不到类的异常  e.printStackTrace();  return false;  }  }   */\*\*  \* 执行增删改操作。  \*  \** ***@param*** *sql 要执行的 SQL 语句  \** ***@param*** *param 参数数组，用于设置预编译 SQL 语句中的占位符  \** ***@return*** *受影响的行数，如果发生错误则返回 -1  \*/* public int executeUpdate(String sql, Object[] param) {  int num = 0;  try {  if (getConnection()) {  // 准备预编译的 SQL 语句  preparedStatement = connection.prepareStatement(sql);   // 设置参数  if (param.length > 0) {  for (int i = 0; i < param.length; i++) {  preparedStatement.setObject((i + 1), param[i]);  }  }   // 执行更新操作  num = preparedStatement.executeUpdate();  }  } catch (SQLException e) {  // 处理 SQL 异常  e.printStackTrace();  return num = -1;  } finally {  // 关闭资源  this.closeResources();  return num;  }  }   */\*\*  \* 执行查询操作。  \*  \** ***@param*** *sql 要执行的 SQL 语句  \** ***@param*** *param 参数数组，用于设置预编译 SQL 语句中的占位符  \** ***@return*** *查询结果集  \*/* public ResultSet executeQuery(String sql, Object[] param) {  try {  if (getConnection()) {  // 准备预编译的 SQL 语句  preparedStatement = connection.prepareStatement(sql);   // 设置参数  if (param.length > 0) {  for (int i = 0; i < param.length; i++) {  preparedStatement.setObject((i + 1), param[i]);  }  }   // 执行查询操作  resultSet = preparedStatement.executeQuery();  }  } catch (SQLException e) {  // 处理 SQL 异常  e.printStackTrace();  } finally {  // 返回结果集  return resultSet;  }  }   */\*\*  \* 关闭所有打开的数据库资源。  \*/* public void closeResources() {  if (preparedStatement != null) {  try {  // 关闭连接和预编译的 SQL 语句对象  connection.close();  preparedStatement.close();  } catch (SQLException e) {  // 抛出运行时异常  throw new RuntimeException(e);  }  }  } }  <%@ **page** import="dao.BaseDao" %><%--  Created by IntelliJ IDEA.  User: Jake  Date: 2024-11-11  Time: 11:02  To change this template use File | Settings | File Templates. --%>  <%@ **page** contentType="text/html;charset=UTF-8" language="java" %> <html> <head>  <title>Title</title> </head> <body> <%  BaseDao baseDao = new BaseDao();  boolean flag = baseDao.getConnection();  if (flag){  out.print("通过专用的JDBC驱动连接Mysql数据库!");  }  else{  out.print("连接失败");  } %> </body> </html> |

3．数据插入操作。本题有两个文件：第一个文件ex5-03.jsp是用户添加信息页面，运行页面如图8所示。单击“添加”按钮，将输入信息提交给ex5-03a.jsp，如果成功，显示如图9所示的成功添加页面，这时打开数据库SCMS，会发现course表中添加了一条记录；否则会输出“记录添加失败！”信息。

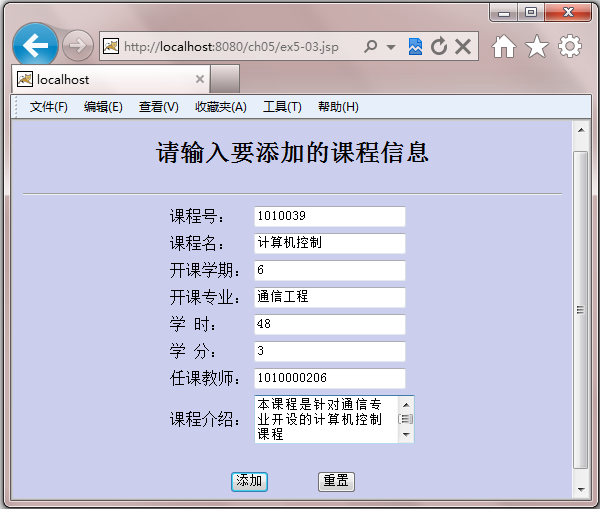


图8 添加课程信息页面

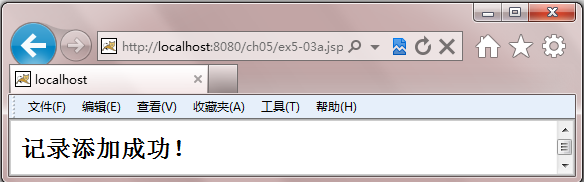
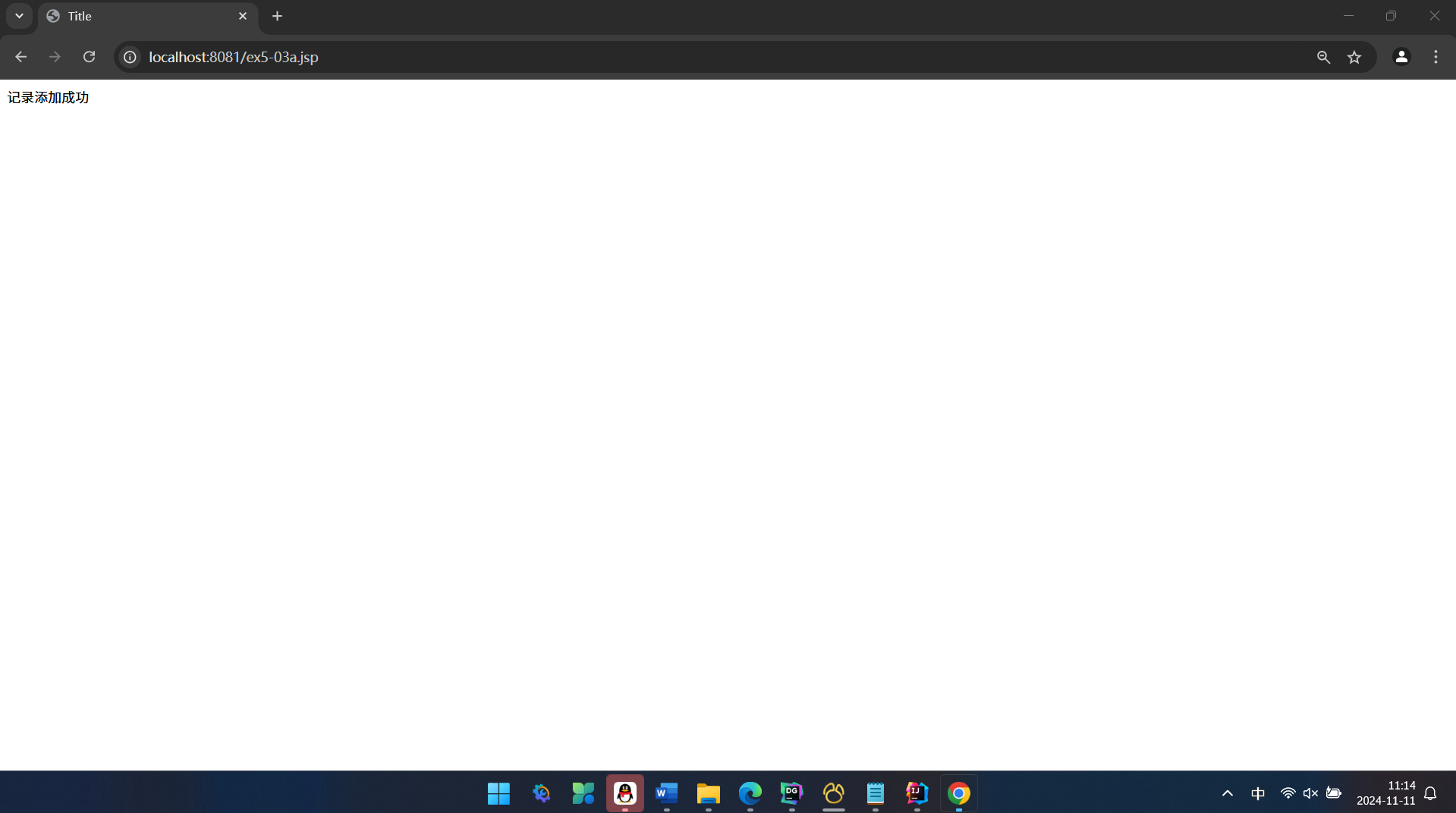


图9 记录添加成功页面



|  |
| --- |
| 此处插入jsp文件代码。  public boolean insert(Course course) {  String sql = "insert into course(cno,cname,cterm,cmajor,chour,ccredit,tno,cintro) values(?,?,?,?,?,?,?,?)";  Object[] param = {course.getCno(),course.getCname(),course.getCterm(),course.getCmajor(),course.getChour(),course.getCcredit(),course.getTno(),course.getCintro()};  int num = super.executeUpdate(sql,param);  if (num > 0){   return true;  }else {  return false;  } }  <%--  Created by IntelliJ IDEA.  User: Jake  Date: 2024-11-11  Time: 10:41  To change this template use File | Settings | File Templates. --%> <%@ **page** contentType="text/html;charset=UTF-8" language="java" %> <html> <head>  <title>添加课程</title> </head> <body> <h2>添加课程信息</h2> <form action="ex5-03a.jsp" method="post">  课程编号: <input type="text" name="cno" required><br>  课程名称: <input type="text" name="cname" required><br>  学期: <input type="number" name="cterm" required><br>  专业: <input type="text" name="cmajor" required><br>  学时: <input type="number" name="chour" required><br>  学分: <input type="number" name="ccredit" required><br>  教师编号: <input type="text" name="tno" required><br>  课程简介: <textarea name="cintro" rows="½" cols="30" required></textarea><br>  <input type="submit" value="添加"> </form> </body> </html>  <%@ **page** import="entity.Course" %> <%@ **page** import="service.impl.CourseService" %><%--  Created by IntelliJ IDEA.  User: Jake  Date: 2024-11-11  Time: 10:41  To change this template use File | Settings | File Templates. --%> <%@ **page** contentType="text/html;charset=UTF-8" language="java" %> <%  request.setCharacterEncoding("UTF-8");  response.setCharacterEncoding("UTF-8");  CourseService courseService = new CourseService();  Course course = new Course();  course.setCno(request.getParameter("cno"));  course.setCname(request.getParameter("cname"));  course.setCterm(Integer.*parseInt*(request.getParameter("cterm")));  course.setCmajor(request.getParameter("cmajor"));  course.setChour(Integer.*parseInt*(request.getParameter("chour")));  course.setCcredit(Integer.*parseInt*(request.getParameter("ccredit")));  course.setTno(request.getParameter("tno"));  course.setCintro(request.getParameter("cintro"));  boolean flag = courseService.insert(course);  if (flag){  out.print("添加成功");  }  else{  out.print("添加失败");  }  %> <html> <head>  <title>Title</title> </head> <body>  </body> </html> |