題目：

寫一支 Java 程式，把 XMLDB 中 class table 資料讀出並寫入 XML 檔，並上傳你的 java 程式碼。

**程式碼**

|  |
| --- |
| import javax.xml.parsers.DocumentBuilderFactory;  import javax.xml.parsers.ParserConfigurationException;  import javax.xml.parsers.DocumentBuilder;  import javax.xml.transform.Transformer;  import javax.xml.transform.TransformerFactory;  import javax.xml.transform.dom.DOMSource;  import javax.xml.transform.stream.StreamResult;  import org.w3c.dom.Attr;  import org.w3c.dom.Document;  import org.w3c.dom.Element;  import java.io.File;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  public class CreateXMLFile  {  static final String DB\_URL = "jdbc:mysql://localhost/XMLDB";  static final String USER = "root";  static final String PASS = "0123456789";  static final String QUERY = "SELECT id, first, last, mark, nick FROM class";  public static void main(String argv[]) throws ParserConfigurationException  {  Integer id;  Integer mark;  String firstn,lastn,nickn;  try  {  DocumentBuilderFactory dbFactory =  DocumentBuilderFactory.newInstance();  DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();  Document doc = dBuilder.newDocument();    // root element  Element rootElement = doc.createElement("class");  doc.appendChild(rootElement);    try(Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);  Statement stmt = conn.createStatement();  ResultSet rs = stmt.executeQuery(QUERY);  )  {  while(rs.next())  {  try  {  //Display values  System.out.print("ID: " + rs.getInt("id"));  id = rs.getInt("id");  System.out.print(", Mark: " + rs.getInt("mark"));  mark = rs.getInt("mark");  System.out.print(", First: " + rs.getString("first"));  firstn = rs.getString("first");  System.out.println(", Last: " + rs.getString("last"));  lastn = rs.getString("last");  System.out.println(", Last: " + rs.getString("nick"));  nickn = rs.getString("nick");    // student element  Element student = doc.createElement("student");  rootElement.appendChild(student);  // setting attribute to element  Attr attr = doc.createAttribute("rollno");  attr.setValue(Integer.toString(id));  student.setAttributeNode(attr);  // firstname element  Element firstname = doc.createElement("firstname");  firstname.appendChild(doc.createTextNode(firstn));  student.appendChild(firstname);    // lastname element  Element lastname = doc.createElement("lastname");  lastname.appendChild(doc.createTextNode(lastn));  student.appendChild(lastname);    // nickname element  Element nickname = doc.createElement("nickname");  nickname.appendChild(doc.createTextNode(nickn));  student.appendChild(nickname);    // marks element  Element marks = doc.createElement("marks");  marks.appendChild(doc.createTextNode(Integer.toString(mark)));  student.appendChild(marks);  }  catch (Exception e)  {  e.printStackTrace();  }  }  }  catch (SQLException e)  {  e.printStackTrace();  }    // write the content into xml file  TransformerFactory transformerFactory = TransformerFactory.newInstance();  Transformer transformer = transformerFactory.newTransformer();  DOMSource source = new DOMSource(doc);    //StreamResult result = new StreamResult(new File("C:\\cars.xml"));  StreamResult result = new StreamResult(new File("test.xml"));  transformer.transform(source, result);    // Output to console for testing  StreamResult consoleResult = new StreamResult(System.out);  transformer.transform(source, consoleResult);  }  catch (Exception e)  {  e.printStackTrace();  }  }  } |

**截圖**

