**Certainly! The provided Java code is part of a simple Hospital Management System, specifically focusing on the Patient class. Let's go through the code line by line with explanations:**

package HospitalManagementSystem;

**This line declares the package name in which the Patient class resides.**

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

**These import statements bring in the necessary classes from the java.sql package for working with JDBC (Java Database Connectivity) and the Scanner class for user input.**

public class Patient {

**This line starts the definition of the Patient class.**

private Connection connection;

private Scanner scanner;

public Patient(Connection connection, Scanner scanner){

this.connection = connection;

this.scanner = scanner;

}

**Here, the class has two private member variables connection and scanner, representing a database connection and a Scanner object for user input, respectively. The constructor Patient is used to initialize these variables when creating an instance of the Patient class.**

public void addPatient(){

**This line starts the definition of the addPatient method, which is used to add a new patient to the database.**

System.out.print("Enter Patient Name: ");

String name = scanner.next();

System.out.print("Enter Patient Age: ");

int age = scanner.nextInt();

System.out.print("Enter Patient Gender: ");

String gender = scanner.next();

**These lines prompt the user to enter the patient's name, age, and gender using the Scanner class for input.**

try{

String query = "INSERT INTO PATIENTS(NAME, AGE, GENDER) VALUES(?,?,?)";

PreparedStatement preparedStatement = connection.prepareStatement(query);

preparedStatement.setString(1, name);

preparedStatement.setInt(2, age);

preparedStatement.setString(3, gender);

int affectedRows = preparedStatement.executeUpdate();

if(affectedRows>0) {

System.out.println("Patient Added Successfully");

}else{

System.out.println("Failed to Add Patient!!");

}

}catch(SQLException e){

e.printStackTrace();

}

}

**In this block, a SQL INSERT query is prepared and executed using a PreparedStatement. The patient's information entered by the user is set as parameters in the prepared statement. The executeUpdate method is used to execute the update (insert) query. If any rows are affected (inserted), a success message is printed; otherwise, a failure message is printed. If there is any SQL exception, it is caught, and the stack trace is printed.**

public void viewPatient(){

**This line starts the definition of the viewPatient method, which is used to display information about all patients in the database.**

String query = "select \* from patients";

try{

PreparedStatement preparedStatement = connection.prepareStatement(query);

ResultSet resultSet = preparedStatement.executeQuery();

System.out.println("Patient: ");

System.out.println("+----------------+-------------------------+------------+----------------+");

System.out.println("| Patient ID | Name | Age | Gender |");

System.out.println("+----------------+-------------------------+------------+-----------------+");

while(resultSet.next()){

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

int age = resultSet.getInt("age");

String gender = resultSet.getString("gender");

System.out.printf("|%-16s|%-25s|%-12s|%-16s|\n",id,name,age,gender);

System.out.println("+----------------+-------------------------+------------+-----------------+");

}

}catch (SQLException e){

e.printStackTrace();

}

}

**In this block, a SQL SELECT query is prepared and executed to retrieve all patient records from the database. The results are then printed in a tabular format using System.out.printf. The while(resultSet.next()) loop iterates through the result set and prints each patient's information.**

public boolean getPatientById(int id){

**This line starts the definition of the getPatientById method, which is used to check if a patient with a specific ID exists in the database.**

String query = "SELECT \* FROM patients WHERE id = ?";

try{

PreparedStatement preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1,id);

ResultSet resultSet = preparedStatement.executeQuery();

if(resultSet.next()) {

return true;

}else{

return false;

}

}catch(SQLException e){

e.printStackTrace();

}

return false;

}

}

**In this block, a SQL SELECT query is prepared and executed to retrieve a patient record based on the specified ID. If a record is found in the result set, the method returns true; otherwise, it returns false. Any SQL exception is caught and printed in the catch block. If an exception occurs, the method also returns false.**