1.A Single ton

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseManager {

private static DatabaseManager instance;

private DatabaseManager() {

try {

Class.forName("org.sqlite.JDBC");

connection = DriverManager.getConnection("jdbc:sqlite:test.db");

} catch (ClassNotFoundException | SQLException e) {

e.printStackTrace();

}

}

public static synchronized DatabaseManager getInstance() {

if (instance == null) {

instance = new DatabaseManager();

}

return instance;

}

public void executeQuery(String query) {

try {

connection.createStatement().execute(query);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void closeConnection() {

try {

if (connection != null) {

connection.close();

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private Connection connection;

public static void main(String[] args) {

DatabaseManager dbManager1 = DatabaseManager.getInstance();

DatabaseManager dbManager2 = DatabaseManager.getInstance();

System.out.println(dbManager1 == dbManager2);

dbManager1.executeQuery("SELECT \* FROM users");

dbManager1.closeConnection();

}

}

2.A Factory Method

public class ShapeFactory {

public Shape createCircle(double radius) {

return new Circle(radius);

}

public Shape createSquare(double sideLength) {

return new Square(sideLength);

}

public Shape createRectangle(double width, double height) {

return new Rectangle(width, height);

}

public static void main(String[] args) {

ShapeFactory factory = new ShapeFactory();

Shape circle = factory.createCircle(5.0);

System.out.println("Circle Area: " + circle.area());

Shape square = factory.createSquare(4.0);

System.out.println("Square Area: " + square.area());

Shape rectangle = factory.createRectangle(3.0, 6.0);

System.out.println("Rectangle Area: " + rectangle.area());

}

}

interface Shape {

double area();

}

class Circle implements Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

return Math.PI \* radius \* radius;

}

}

class Square implements Shape {

private double sideLength;

public Square(double sideLength) {

this.sideLength = sideLength;

}

@Override

public double area() {

return sideLength \* sideLength;

}

}

class Rectangle implements Shape {

private double width;

private double height;

public Rectangle(double width, double height) {

this.width = width;

this.height = height;

}

@Override

public double area() {

return width \* height;

}

}