Câu 1; Thực hiện ứng dụng bên mày chủ

a.Tạo lớp quản lý thông tin sản phẩm

b. Tạo lớp thực hiện chức năng nhận và gửi dữ liệu

Câu 2: Thực hiện ứng dụng bên máy khách

a.Tạo Form có chức năg tìm kiếm thông tin sản phẩm

b.tạo lớp thực hiện chức năng gửi và nhận dữ liệu

Câu1a

Implement.java’

package rmi;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class Implement extends UnicastRemoteObject implements InterfacePro {

private static final long serialVersionUID = 1L;

private Connection connection;

public Implement() throws RemoteException {

super();

// Khởi tạo kết nối JDBC ở đây

try {

// Thiết lập kết nối JDBC

String url = "jdbc:mySql://localhost:3306/product\_management";

String username = "root";

String password = "1234";

connection = DriverManager.getConnection(url, username, password);

} catch (SQLException e) {

e.printStackTrace();

}

}

// Các phương thức được triển khai từ InterfacePro

// getAllProducts(), getProductById(), addProduct(), updateProduct(), deleteProduct()

}

b. Tạo lớp thực hiện chức năng nhận và gửi dữ liệu:

server.java

package rmi;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

public class Server {

public static void main(String[] args) {

try {

Implement productManager = new Implement();

Registry registry = LocateRegistry.createRegistry(1099);

registry.rebind("ProductManager", productManager);

System.out.println("Server started.");

} catch (Exception e) {

System.err.println("Server exception: " + e.toString());

e.printStackTrace();

}

}

}

Câu 2: Thực hiện ứng dụng bên máy khách

1. Tạo Form có chức năng tìm kiếm thông tin sản phẩm:

Client.java

package rmi;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class ClientForm extends JFrame {

private InterfacePro productManager;

public ClientForm(InterfacePro productManager) {

this.productManager = productManager;

initComponents();

}

private void initComponents() {

setTitle("Product Search");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 300);

setLocationRelativeTo(null);

JPanel panel = new JPanel();

JLabel titleLabel = new JLabel("Search Product by ID:");

panel.add(titleLabel);

JTextField idField = new JTextField(10);

panel.add(idField);

JButton searchButton = new JButton("Search");

searchButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int productId = Integer.parseInt(idField.getText());

searchProductById(productId);

}

});

panel.add(searchButton);

getContentPane().add(panel);

}

private void searchProductById(int productId) {

try {

Product product = productManager.getProductById(productId);

if (product != null) {

JOptionPane.showMessageDialog(this, "Product with ID " + productId + ":\n" + product.toString(), "Search Result", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(this, "No product found with ID " + productId + ".", "Search Result", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Error searching for product: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

}

B,File ClientDataHandler.java (Tạo lớp thực hiện chức năng gửi và nhận dữ liệu)

package rmi;

import javax.swing.\*;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

public static void main(String[] args) {

try {

Registry registry = LocateRegistry.getRegistry("localhost", 1099);

InterfacePro productManager = (InterfacePro) registry.lookup("ProductManager");

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new Client(productManager).setVisible(true);

}

});

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "Client exception: " + e.toString(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

conletciton sản phẩm

import java.util.ArrayList;

import java.util.List;

public class ProductCollectionExample {

public static void main(String[] args) {

// Tạo một danh sách sản phẩm (ArrayList)

List<Product> productList = new ArrayList<>();

// Thêm các sản phẩm vào danh sách

productList.add(new Product(1, "Laptop", 999.99));

productList.add(new Product(2, "Smartphone", 599.99));

productList.add(new Product(3, "Tablet", 299.99));

// Hiển thị thông tin về các sản phẩm trong danh sách

System.out.println("Danh sách sản phẩm:");

for (Product product : productList) {

System.out.println(product);

}

}

}

// Định nghĩa lớp Product

class Product {

private int id;

private String name;

private double price;

// Constructor

public Product(int id, String name, double price) {

this.id = id;

this.name = name;

this.price = price;

}

// Phương thức toString() để hiển thị thông tin về sản phẩm

@Override

public String toString() {

return "Product{" +

"id=" + id +

", name='" + name + '\'' +

", price=" + price +

'}';

}

}

Hoặc  
class Product {

private int id;

private String name;

private double price;

// Constructor

public Product(int id, String name, double price) {

this.id = id;

this.name = name;

this.price = price;

}

// Phương thức getter cho id

public int getId() {

return id;

}

// Phương thức setter cho id

public void setId(int id) {

this.id = id;

}

// Phương thức getter cho name

public String getName() {

return name;

}

// Phương thức setter cho name

public void setName(String name) {

this.name = name;

}

// Phương thức getter cho price

public double getPrice() {

return price;

}

// Phương thức setter cho price

public void setPrice(double price) {

this.price = price;

}

// Phương thức toString() để hiển thị thông tin về sản phẩm

@Override

public String toString() {

return "Product{" +

"id=" + id +

", name='" + name + '\'' +

", price=" + price +

'}';

}

}