МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

ЗВІТ

з лабораторної роботи № 2

з дисципліни

«ПРОГРАМУВАННЯ ВЕБ-ЗАСТОСУВАНЬ JAVA EE»

ВИКОНАВ

Студент групи КН-36а

Кулик В.В.

ПЕРЕВІРИВ

Доцент каф. ПІІТУ

Бабкова Н. В.

Харків 2019

Тема

«ДОСЛІДЖЕННЯ ПРИНЦИПІВ СТВОРЕННЯ ВЕБ-ДОДАТКУ ДЛЯ МАНІПУЛЮВАННя ДАНИМИ JAVA»

Завдання роботи

1. Обрати предметну область для виконання лабораторних робіт.
2. Розробити та реалізувати структуру класів. Створити колекції об’єктів із створених класів та реалізувати обробку кожної з них.
3. Реалізувати веб-застосування, що створюється за технологією JSP: main, master, detail (другу та третю назвати відповідно до обраної предметної області). Реалізувати виклик сторінок. Реалізувати повернення на сторінку main.
4. Сторінка main формується на основі класу-колекції екземпляру головного класу. Реалізувати додавання, редагування та вилучення записів. Вивантажити перед виходом у xml-файл.
5. Сторінка detail формується на основі колекції detail. Реалізувати фільтрацію даних.

Хід роботи

1. Була обрана предметна область “Магазин комп’ютерних комплектуючих”. Маємо три сутності (класи): Product – Orders – Providers. Веб-застосунок дозволяє працівникам магазину контролювати замовлення від постачальника. У замовленнях відображається назва товару, його постачальник, кількість та остаточна вартість всього замовлення. Інтерфейс дозволяє редагувати та фільтрувати дані.

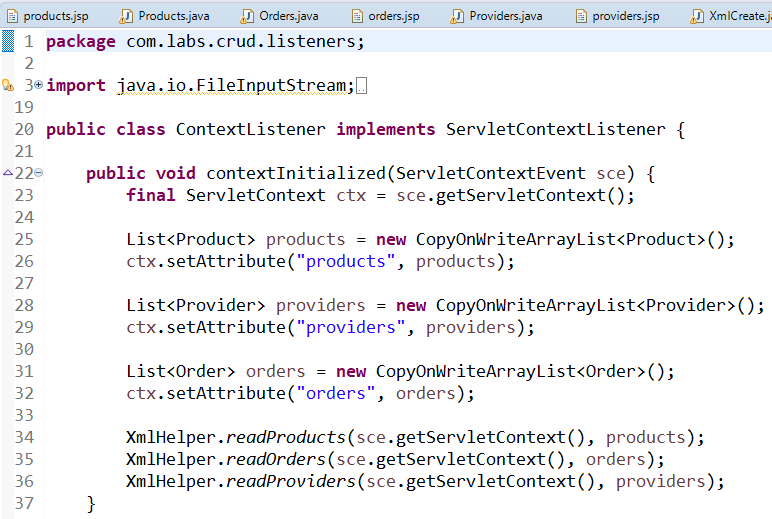
Створюємо колекції даних для класів Product, Order, Provider. 

Рисунок 1 – Колекції даних

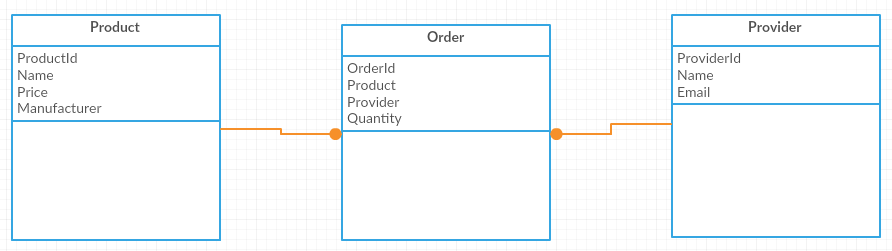


Рисунок 2 – Діаграма класів

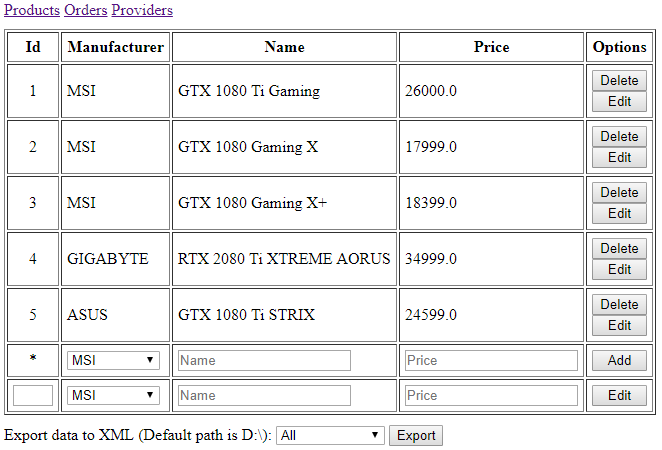


Рисунок 3 – Головна сторінка товарів

Код JSP головної сторінки Products:

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Products</title>

<link rel="stylesheet" href="styles/main.css">

</head>

<body>

<div class="navigator">

<a href="/crud">Products</a>

<a href="Orders">Orders</a>

<a href="Providers">Providers</a>

</div>

<table border="1" cellpadding="5">

<tr>

<th>Id</th>

<th>Manufacturer</th>

<th>Name</th>

<th>Price</th>

<th>Options</th>

</tr>

<c:forEach var = "product" items="${requestScope.products}">

<tr>

<td class="id">${product.id}</td>

<td>${product.manufacturer}</td>

<td>${product.name}</td>

<td>${product.price}</td>

<td>

<form method="POST" action="ProductDelete">

<input hidden name="product" value="${product.id}">

<input type="submit" value="Delete">

</form>

<form method="GET" action="ProductEdit">

<input hidden name="product" value="${product.id}">

<input type="submit" value="Edit">

</form>

</td>

</tr>

</c:forEach>

<tr>

<form method="POST" action="ProductCreate">

<td class="id">\*</td>

<td><select name="manufacturer">

<option value="MSI">MSI</option>

<option value="ASUS">ASUS</option>

<option value="GIGABYTE">GIGABYTE</option>

<option value="PALIT">PALIT</option>

<option value="NVIDIA ref.">NVIDIA ref.</option>

<option value="EVGA">EVGA</option>

<option value="ZOTAC">ZOTAC</option>

</select></td>

<td><input type="text" name="name" placeholder="Name"></td>

<td><input type="text" name="price" placeholder="Price"></td>

<td><input type="submit" value="Add"></td>

</form>

</tr>

<tr>

<form method="POST" action="ProductEdit">

<td class="id"><input type="text" readonly size="1" name="product" value="${requestScope.product.id}"></td>

<td><select name="manufacturer">

<option value="MSI">MSI</option>

<option value="ASUS">ASUS</option>

<option value="GIGABYTE">GIGABYTE</option>

<option value="PALIT">PALIT</option>

<option value="NVIDIA ref.">NVIDIA ref.</option>

<option value="EVGA">EVGA</option>

<option value="ZOTAC">ZOTAC</option>

</select></td>

<td><input type="text" name="name" placeholder="Name" value="${requestScope.product.name}"></td>

<td><input type="text" name="price" placeholder="Price" value="${requestScope.product.price}"></td>

<td><input type="submit" value="Edit"></td>

</form>

</tr>

</table>

<form action="XmlCreate" method="POST">

<lable for="exportBtn">Export data to XML (Default path is D:\):</lable>

<select name="type">

<option value="all">All</option>

<option value="products">Products only</option>

<option value="orders">Orders only</option>

<option value="providers">Providers only</option>

</select>

<input id="exportBtn" type="submit" value="Export">

</form>

</body>

</html>

Код сервлету Products, який відповідає за основну логіку (додавання, видалення, редагування):

package com.labs.crud.servlets;

import java.beans.XMLDecoder;

import java.io.\*;

import java.util.\*;

import java.util.concurrent.CopyOnWriteArrayList;

import java.util.concurrent.atomic.AtomicInteger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.xml.sax.SAXException;

import com.labs.crud.classes.Product;

import com.labs.crud.util.XmlHelper;

public class Products extends HttpServlet {

private static final long serialVersionUID = 1L;

private List<Product> products;

public Products() {

super();

}

public void init(){

final Object products = getServletContext().getAttribute("products");

if (products == null || !(products instanceof CopyOnWriteArrayList)) {

throw new IllegalStateException();

} else {

this.products = (CopyOnWriteArrayList<Product>) products;

}

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setAttribute("products", products);

request.getRequestDispatcher("products.jsp").forward(request, response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

}

}

Сторінка Orders на якій реалізовано фільтрацію даних за назвою постачальника:

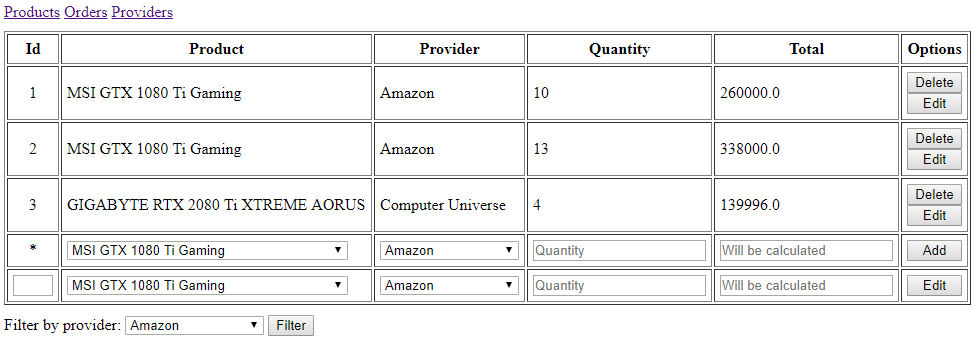


Рисунок 4 – Сторінка замовлень

Код сервлету Orders:

package com.labs.crud.servlets;

import java.beans.XMLDecoder;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.\*;

import java.util.concurrent.CopyOnWriteArrayList;

import java.util.concurrent.atomic.AtomicInteger;

import java.util.stream.IntStream;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

import com.labs.crud.classes.Order;

import com.labs.crud.classes.Product;

import com.labs.crud.classes.Provider;

import com.labs.crud.util.XmlHelper;

public class Orders extends HttpServlet {

private static final long serialVersionUID = 1L;

private List<Order> orders;

public Orders() {

super();

}

public void init() {

final Object orders = getServletContext().getAttribute("orders");

if (orders == null || !(orders instanceof CopyOnWriteArrayList)) {

throw new IllegalStateException();

} else {

this.orders = (CopyOnWriteArrayList<Order>) orders;

}

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setAttribute("orders", orders);

request.getRequestDispatcher("orders.jsp").forward(request, response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

if (request.getParameter("provider") != null) {

String provider = request.getParameter("provider");

List<Order> orders = new CopyOnWriteArrayList<Order>();

for (Order o : this.orders) {

if(o.getProvider().getName().equals(provider)) {

orders.add(o);

}

}

request.setAttribute("orders", orders);

request.getRequestDispatcher("/orders.jsp").forward(request, response);

}

}

}

Нижче представленний код сторінки JSP:

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Orders</title>

<link rel="stylesheet" href="styles/main.css">

</head>

<body>

<div class="navigator">

<a href="/crud">Products</a>

<a href="Orders">Orders</a>

<a href="Providers">Providers</a>

</div>

<table border="1" cellpadding="5">

<tr>

<th>Id</th>

<th>Product</th>

<th>Provider</th>

<th>Quantity</th>

<th>Total</th>

<th>Options</th>

</tr>

<c:forEach var = "order" items="${requestScope.orders}">

<tr>

<td class="id">${order.id}</td>

<td>${order.product.manufacturer} ${order.product.name}</td>

<td>${order.provider.name}</td>

<td>${order.quantity}</td>

<td>${order.quantity \* order.product.price}</td>

<td>

<form method="POST" action="OrderDelete">

<input hidden name="order" value="${order.id}">

<input type="submit" value="Delete">

</form>

<form method="GET" action="OrderEdit">

<input hidden name="order" value="${order.id}">

<input type="submit" value="Edit">

</form>

</td>

</tr>

</c:forEach>

<tr>

<form method="POST" action="OrderCreate">

<td class="id">\*</td>

<td><select name="product">

<c:forEach var = "product" items="${applicationScope.products}">

<option value="${product.id}">${product.manufacturer} ${product.name}</option>

</c:forEach>

</select></td>

<td><select name="provider">

<c:forEach var = "provider" items="${applicationScope.providers}">

<option value="${provider.id}">${provider.name}</option>

</c:forEach>

</select></td>

<td><input type="text" name="quantity" placeholder="Quantity"></td>

<td><input type="text" readonly placeholder="Will be calculated"></td>

<td><input type="submit" value="Add"></td>

</form>

</tr>

<tr>

<form method="POST" action="OrderEdit">

<td class="id"><input type="text" readonly size="1" name="order" value="${requestScope.order.id}"></td>

<td><select name="product">

<c:forEach var = "product" items="${applicationScope.products}">

<option value="${product.id}">${product.manufacturer} ${product.name}</option>

</c:forEach>

</select></td>

<td><select name="provider">

<c:forEach var = "provider" items="${applicationScope.providers}">

<option value="${provider.id}">${provider.name}</option>

</c:forEach>

</select></td>

<td><input type="text" name="quantity" value="${requestScope.order.quantity }" placeholder="Quantity"></td>

<td><input type="text" readonly placeholder="Will be calculated"></td>

<td><input type="submit" value="Edit"></td>

</form>

</tr>

</table>

<form action="Orders" method="POST">

<lable for="providerPico">Filter by provider:</lable>

<select id="providerPico" name="provider">

<c:forEach var = "provider" items="${applicationScope.providers}">

<option value="${provider.name}">${provider.name}</option>

</c:forEach>

</select>

<input type="submit" value="Filter">

</form>

</body>

</html>

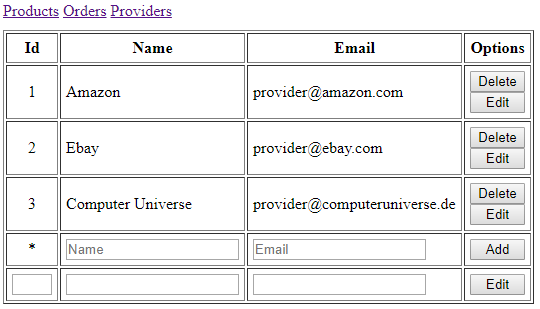


Рисунок 5 – Таблиця постачальників

Код відповідного сервлету:

package com.labs.crud.servlets;

import java.beans.XMLDecoder;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.\*;

import java.util.concurrent.CopyOnWriteArrayList;

import java.util.concurrent.atomic.AtomicInteger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.xml.sax.SAXException;

import com.labs.crud.classes.Product;

import com.labs.crud.classes.Provider;

import com.labs.crud.util.XmlHelper;

public class Providers extends HttpServlet {

private static final long serialVersionUID = 1L;

private List<Provider> providers;

public Providers() {

super();

}

public void init() {

final Object providers = getServletContext().getAttribute("providers");

final String path = getServletContext().getInitParameter("xmlPath") + getServletContext().getInitParameter("providersPath");

if (providers == null || !(providers instanceof CopyOnWriteArrayList)) {

throw new IllegalStateException();

} else {

this.providers = (CopyOnWriteArrayList<Provider>) providers;

}

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setAttribute("providers", providers);

request.getRequestDispatcher("providers.jsp").forward(request, response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

}

}

Код JSP сторінки постачальників:

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Providers</title>

<link rel="stylesheet" href="styles/main.css">

</head>

<body>

<div class="navigator">

<a href="/crud">Products</a>

<a href="Orders">Orders</a>

<a href="Providers">Providers</a>

</div>

<table border="1" cellpadding="5">

<tr>

<th>Id</th>

<th>Name</th>

<th>Email</th>

<th>Options</th>

</tr>

<c:forEach var = "provider" items="${requestScope.providers}">

<tr>

<td class="id">${provider.id}</td>

<td>${provider.name}</td>

<td>${provider.email}</td>

<td>

<form method="POST" action="ProviderDelete">

<input hidden name="provider" value="${provider.id}">

<input type="submit" value="Delete">

</form>

<form method="GET" action="ProviderEdit">

<input hidden name="provider" value="${provider.id}">

<input type="submit" value="Edit">

</form>

</td>

</tr>

</c:forEach>

<tr>

<form method="POST" action="ProviderCreate">

<td class="id">\*</td>

<td><input type="text" name="name" placeholder="Name"></td>

<td><input type="email" name="email" placeholder="Email"></td>

<td><input type="submit" value="Add"></td>

</form>

</tr>

<tr>

<form method="POST" action="ProviderEdit">

<td class="id"><input type="text" readonly size="1" name="provider" value="${requestScope.provider.id}"></td>

<td><input type="text" name="name" value="${requestScope.provider.name}"></td>

<td><input type="email" name="email" value="${requestScope.provider.email}"></td>

<td><input type="submit" value="Edit"></td>

</form>

</tr>

</table>

</body>

</html>

Код сервлету, який дозволяє вивантажити колекцію об'єктів у файл XML:

package com.labs.crud.servlets;

import java.io.IOException;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.concurrent.CopyOnWriteArrayList;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.labs.crud.classes.Order;

import com.labs.crud.classes.Product;

import com.labs.crud.classes.Provider;

import com.labs.crud.util.XmlHelper;

/\*\*

\* Servlet implementation class XmlCreate

\*/

public class XmlCreate extends HttpServlet {

private static final long serialVersionUID = 1L;

private List<Order> orders;

private List<Product> products;

private List<Provider> providers;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public XmlCreate() {

super();

// TODO Auto-generated constructor stub

}

public void init() {

final Object orders = getServletContext().getAttribute("orders");

final Object products = getServletContext().getAttribute("products");

final Object providers = getServletContext().getAttribute("providers");

if (orders == null || !(orders instanceof CopyOnWriteArrayList)) {

throw new IllegalStateException();

} else {

this.orders = (CopyOnWriteArrayList<Order>) orders;

}

if (products == null || !(products instanceof CopyOnWriteArrayList)) {

throw new IllegalStateException();

} else {

this.products = (CopyOnWriteArrayList<Product>) products;

}

if (providers == null || !(providers instanceof CopyOnWriteArrayList)) {

throw new IllegalStateException();

} else {

this.providers = (CopyOnWriteArrayList<Provider>) providers;

}

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String path = getServletConfig().getInitParameter("xmlPath");

String ordersPath = path + getServletContext().getInitParameter("ordersPath");

String productsPath = path + getServletContext().getInitParameter("productsPath");

String providersPath = path + getServletContext().getInitParameter("providersPath");

switch(request.getParameter("type")) {

case "all":

XmlHelper.exportOrders(this.orders, ordersPath);

XmlHelper.exportProducts(this.products, productsPath);

XmlHelper.exportProviders(this.providers, providersPath);

break;

case "products":

XmlHelper.getProductXml(this.products, response);

break;

case "providers":

XmlHelper.getProviderXml(this.providers, response);

break;

case "orders":

XmlHelper.getOrderXml(this.orders, response);

break;

}

response.sendRedirect("./");

}

}



Рисунок 6 – вивантажений XML

**Висновок**: на даній лабораторній роботі було досліджено стек технології для створення веб-додатку на java, створення Servlet та jsp файлів, вивантаження колекції об’єктів у XML-файл, роботу з колекціями (додавання, видалення, редагування).