



Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

ЛАБОРАТОРНА РОБОТА №4

з дисципліни «Основи програмування - 2»

Тема: «EL, JSTL»

Виконали:

студенти групи ІА-31
Клим'юк В.Л, Самелюк А.С,
Дук М.Д, Сакун Д.С

Перевірив:

асистент кафедри ІСТ
Степанов А. С.

Тема: EL, JSTL

Мета: Мета лабораторної роботи полягає в оволодінні основними концепціями та функціями двох популярних технологій веб-розробки - Expression Language (EL) та JavaServer Pages Standard Tag Library (JSTL). Ця робота спрямована на розуміння використання EL для звернення до об'єктів Java та виконання виразів в JSP, а також на використання різноманітних тегів JSTL для виконання рутинних операцій, таких як управління потоком, обробка колекцій даних та форматування виводу.

Хід роботи

1. Пригадати як використовувати JSP, EL, JSTL для створення динамічних веб-сторінок.
2. Для свого варіанту з л/р №10 першого семестру створити web-застосунок використовуючи шаблон MVC:
 - модель (Model) має містити дані;
 - контролер (Controller) має забезпечувати навігацію між сторінками та здійснювати виклики до бізнес-логіки, отримувати від неї результати, та передавати їх у представлення для відображення;
 - представлення (View) має відображати результати та містити елементи навігації.

Java-код має бути присутнім лише у моделі та контролерах. JSP-сторінки потрібно реалізувати без використання Java, а використовуючи лише EL та JSTL. В процесі виконання завдання потрібно продемонструвати використання таких JSTL-елементів: <c:out ... >, <c:if ... >, <c:forEach ... >.

```
package com.example.lab_04;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;

@WebServlet(name = "indexServlet", value = "/")
public class IndexServlet extends HttpServlet {
```

```

    public void init() {
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException {
        request.getRequestDispatcher("main.jsp").forward(request, response);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException {
        String action = request.getParameter("action");
        request.setAttribute("action", action);

        switch (action){
            case "all_u":{
                var res = MainService.countAllStudents();
                request.setAttribute("res", res);
                break;
            }
            case "most":{
                var res = MainService.findFacultyWithMostStudents();
                request.setAttribute("res", res);
                break;
            }
            case "all_g":{
                var res = MainService.findStudentsWithGrade();
                request.setAttribute("res", res);
                break;
            }
        }

        request.getRequestDispatcher("main.jsp").forward(request, response);
    }

    public void destroy() {
    }
}

```

Код 1.1 – IndexServlet

```

package com.example.lab_04;

import com.example.lab_04.Classes.Faculty;
import com.example.lab_04.Classes.Student;
import com.example.lab_04.Classes.University;

import java.util.*;

public class MainService {
    private final static University university;

    static{
        Student s1 = new Student("name1", "surname1", 1, 90);
        Student s2 = new Student("name2", "surname2", 2, 20);
        Student s3 = new Student("name3", "surname3", 3, 30);
        Student s4 = new Student("name4", "surname4", 4, 40);
        Student s5 = new Student("name5", "surname5", 5, 50);
        Student s6 = new Student("name6", "surname6", 6, 60);
        Student s7 = new Student("name7", "surname7", 7, 70);
        Student s8 = new Student("name8", "surname8", 8, 80);
        Student s9 = new Student("name9", "surname9", 9, 95);
    }
}

```

```

Student s10 = new Student("name10", "surname10", 10, 100);
Student s11 = new Student("name11", "surname11", 11, 98);
Student s12 = new Student("name12", "surname12", 12, 50);
Student s13 = new Student("name13", "surname13", 13, 50);

ArrayList<Student> l1 = new ArrayList<>();
ArrayList<Student> l2 = new ArrayList<>();
ArrayList<Student> l3 = new ArrayList<>();
ArrayList<Student> l4 = new ArrayList<>();
ArrayList<Student> l5 = new ArrayList<>();

l1.add(s1); l1.add(s3); l1.add(s4); l1.add(s8);
l2.add(s10); l2.add(s5);
l3.add(s13); l3.add(s2); l3.add(s7); l3.add(s9); l3.add(s11);
l5.add(s6); l5.add(s12);

Faculty f1 = new Faculty("Faculty1", l1);
Faculty f2 = new Faculty("Faculty2", l2);
Faculty f3 = new Faculty("Faculty3", l3);
Faculty f4 = new Faculty("Faculty4", l4);
Faculty f5 = new Faculty("Faculty5", l5);

HashMap<String, Faculty> faculties = new HashMap<>();

faculties.put(f1.get_name(), f1);
faculties.put(f2.get_name(), f2);
faculties.put(f3.get_name(), f3);
faculties.put(f4.get_name(), f4);
faculties.put(f5.get_name(), f5);

university = new University("University1", faculties);
}

public static int countAllStudents(){
    return university.countAllStudents();
}

public static String findFacultyWithMostStudents(){
    return university.findFacultyWithMostStudents();
}

public static ArrayList<Student> findStudentsWithGrade(){
    return university.findStudentsWithGrade();
}
}

```

Код 1.2 – MainServlet

```

package com.example.lab_04.Classes;

import java.util.ArrayList;
import java.util.Iterator;

public class Faculty {
    public String get_name() {
        return _name;
    }

    public void set_name(String _name) {
        this._name = _name;
    }
}

```

```

private String _name;
private ArrayList<Student> _students = new ArrayList<>();

public Faculty(String name, ArrayList<Student> students){
    this._name = name;
    this._students = students;
}

public void addStudent(Student s){
    _students.add(s);
}

public int countAllStudents(){
    int res = 0;
    Iterator itr = _students.iterator();
    while(itr.hasNext()){
        res++;
        itr.next();
    }
    return res;
}

public ArrayList<Student> findStudentsWithGrade(){
    ArrayList<Student> res = new ArrayList<>();

    for(Student s : _students){
        int g = s.get_grade();
        if (g >= 95 && g <= 100){
            res.add(s);
        }
    }

    return res;
}
}

```

Код 1.3 – Faculty

```

package com.example.lab_04.Classes;

public class Student {
    public String get_name() {
        return _name;
    }

    public void set_name(String _name) {
        this._name = _name;
    }

    public String get_surname() {
        return _surname;
    }

    public void set_surname(String _surname) {
        this._surname = _surname;
    }

    public int get_number() {
        return _number;
    }

    public void set_number(int _number) {

```

```

        this._number = _number;
    }

    public int get_grade() {
        return _grade;
    }

    public void set_grade(int _grade) {
        this._grade = _grade;
    }

    private String _name;
    private String _surname;
    private int _number;
    private int _grade;

    public Student(String name, String surname, int number, int grade){
        this._name = name;
        this._surname = surname;
        this._number = number;
        this._grade = grade;
    }
}

```

Код 1.4 – Student

```

package com.example.lab_04.Classes;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;

public class University {
    public String get_name() {
        return _name;
    }

    public void set_name(String _name) {
        this._name = _name;
    }

    public University(String name, HashMap<String, Faculty> faculties){
        this._name = name;
        this._faculties = faculties;
    }

    private String _name = "University1";

    private HashMap<String, Faculty> _faculties = new HashMap<>();

    public void addFaculty(Faculty faculty){
        if (_faculties.containsKey(faculty.get_name())){
            throw new RuntimeException("Faculty is already present in hashmap");
        }
        else{
            _faculties.put(faculty.get_name(), faculty);
        }
    }

    public int countAllStudents(){
        int res = 0;
        Iterator itr = _faculties.keySet().iterator();
    }
}

```

```

        while(itr.hasNext()){
            res += _faculties.get(itr.next()).countAllStudents();
        }
        return res;
    }

    public String findFacultyWithMostStudents(){
        Faculty res = null;
        Iterator<String> itr = _faculties.keySet().iterator();
        while(itr.hasNext()){
            if (res == null){
                res = _faculties.get(itr.next());
            }
            else{
                Faculty c = _faculties.get(itr.next());
                if (c.countAllStudents() > res.countAllStudents()){
                    res = c;
                }
            }
        }

        if (res == null){
            throw new RuntimeException("University has no faculties");
        }
        return res.get_name();
    }

    public ArrayList<Student> findStudentsWithGrade(){
        ArrayList<Student> res = new ArrayList<>();

        for(String name : _faculties.keySet()){
            res.addAll(_faculties.get(name).findStudentsWithGrade());
        }

        return res;
    }
}

```

Код 1.5 – University

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
    <title>JSP - Hello World</title>
    <link rel="stylesheet" href="css/main.css">
</head>
<body>
    <jsp:include page="header.jsp"/>
    <main>
        <form method="post">
            <div>Count all students in university</div>
            <input type="hidden" name="action" value="all_u">
            <button type="submit">Count</button>
        </form>
        <c:if test="${requestScope.action == 'all_u'}">
            <div>
                <c:out value="Result: ${requestScope.res}">
                </c:out>
            </div>
        </c:if>
    </main>
</body>

```

```

</c:if>
<form method="post">
    <div>Find faculty with most students</div>
    <input type="hidden" name="action" value="most">
    <button type="submit">Find</button>
</form>
<c:if test="${requestScope.action == 'most'}">
    <div>
        <c:out value="Result: ${requestScope.res}">
        </c:out>
    </div>
</c:if>
<form method="post">
    <div>Count all students with 95-100 grade</div>
    <input type="hidden" name="action" value="all_g">
    <button type="submit">Count</button>
</form>
<c:if test="${requestScope.action == 'all_g'}">
    <div>
        <c:out value="Results:">
        </c:out>
        <div>
            <c:forEach var="i" begin="0" end="${requestScope.res.size() -
1}">
                <div>
                    <span>
                        <c:out value="Name:
${requestScope.res.get(i).get_name()} ">
                        </c:out>
                    </span>
                    <span>
                        <c:out value=" Surname:
${requestScope.res.get(i).get_surname()} ">
                        </c:out>
                    </span>
                    <span>
                        <c:out value=" Number:
${requestScope.res.get(i).get_number()} ">
                        </c:out>
                    </span>
                    <span>
                        <c:out value=" Grade:
${requestScope.res.get(i).get_grade()} ">
                        </c:out>
                    </span>
                </div>
            </c:forEach>
        </div>
    </div>
</c:if>
</main>
</body>
</html>

```

Код 1.6 – main.jsp

```

<header>
    <div>Header</div>
</header>

```

Код 1.7 – header.jsp

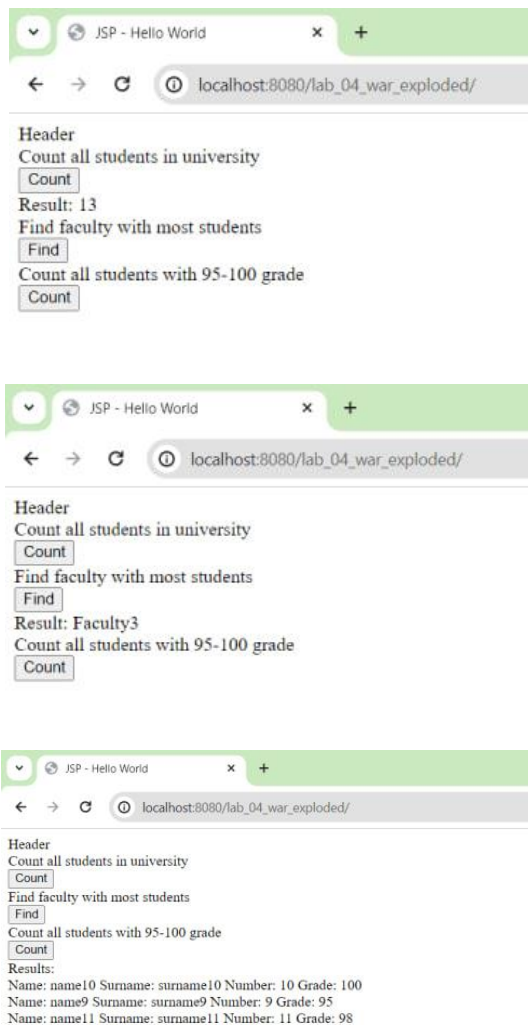


Рисунок 1.(1-3) – Приклади виконання роботи

3. Відповісти на контрольні питання.

Висновки: У висновку хочеться відзначити, що вивчення EL та JSTL дозволило отримати цінні навички для розробки динамічних веб-сторінок на базі Java EE. Застосування EL дозволяє зробити код більш зрозумілим та компактним, а використання JSTL спрощує роботу з даними та зменшує кількість коду, необхідного для виконання різних завдань. Таким чином, отримані знання виявляються важливими для ефективного розвитку сучасних веб-додатків.