

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА № 4

з дисципліни «Основи програмування»

Тема: EL, JSTL

Виконали:

Студентки групи ІА-31

Горlach Д., Макасеєва М.,

Соколова П.

Перевірів:

Степанов А.

Тема: EL, JSTL

Мета: Ознайомитись з наданими нам матеріалами, пригадати інформацію надану на лекціях з даної теми, також навчившись з минулих лабораторних робіт правильно використовувати знання та реалізовувати за допомогою них завдання, виконати поставлену нам задачу. А саме, скористатись новими знаннями та реалізувати за допомогою них подане завдання.

Хід роботи:

Для виконання роботи ми повторили хід роботи попередніх лабораторних робіт: Спочатку ми додаємо в рtm всі потрібні залежності для того, щоб працювати з сервлетами. Створюємо папку webapp і налаштовуємо Tomcat. Модель це класи Comment, Видео і VideoBlog, і HarryPotterVideoBlog це бізнес логіка де ми отримуємо дані про всі моделі з попередньої лабораторної роботи No10 (відео блог...). Сервлети (CommentIsGreaterThanVideoLikes, VideoTotalWatchServlet, VideoWithMostDislikesServlet) це контролери які отримують бізнес логіку і передають у JSP сторінки (View). В JSP сторінках відображаємо моделі за допомогою EL (Expression Language) і JSTL(Jakarta Standard Tag Library). В JSP точніше використовуємо EL і JSTL для відображення логіки і з лабораторної роботи No10. Також на головній сторінці (index.html) є навігація на інші сторінки.

Основи програмування - 2

Лабораторна робота № 4

Тема: EL, JSTL

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 ... >.

3. Відповісти на контрольні питання

Рис 1. Задача

Основи програмування
Лабораторна робота №10
Тема: Колекції. Списки

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів та класів:

- Collection
- List
- ArrayList
- LinkedList
- Iterator
- RandomAccess

2. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці 1. Для цього:

- проаналізувати завдання;
- створити зазначенні класи;
- для створення списків слід використовувати класи та інтерфейси з Collection Framework (заборонено використовувати масиви);
- усі списки мають бути типізованими (наприклад, ArrayList<Student>, а не просто ArrayList);
- при реалізації задач «1)», «2)», «3)» слід застосувати наступні методи перегляду колекцій у відповідності до свого варіанту (табл. 1):
 - a) нетипізований ітератор;
 - b) типізований ітератор;
 - c) типізований цикл «for-each».

Рис 2. Завдання лаб.10 з минулого семестру

Таблиця 1. Таблиця варіантів

Варіант	Завдання	Тип ітератора		
		Задача 1	Задача 2	Задача 3
1	1	a	b	c
2	2	a	c	b
3	3	b	a	c
4	4	b	c	a
5	5	c	a	b
6	1	c	b	a
7	2	a	b	c
8	3	a	c	b
9	4	b	a	c
10	5	b	c	a
11	1	c	a	b
12	2	c	b	a
13	3	a	b	c
14	4	a	c	b
15	5	b	a	c
16	1	b	c	a
17	2	c	a	b

Рис. 3 вар. 9 - наш варіант лаб. 10 першого семестру

4	<p>Класи: Відеоблог (відеоблогер, список відео) Відео (назва, URL, кількість переглядів, кількість лайків, кількість дізлайків, список коментарів) Коментар (текст, кількість лайків, кількість дізлайків)</p> <p>Задача: 1) Знайти загальну сумарну кількість переглядів усіх відео разом 2) Визначити, чи є коментар, що набрав лайків більше, ніж відео, до якого він дописаний 3) Визначити відео, що набрало найбільшу кількість дізлайків. Якщо таких відео декілька, то скласти їх список. Якщо таке відео одне, цей список повинен мати один елемент. Якщо таких відео немає, список має бути порожнім (без елементів)</p>
---	--

Рис. 4 наш варіант лаб. 10 1 семестру - videoBlog

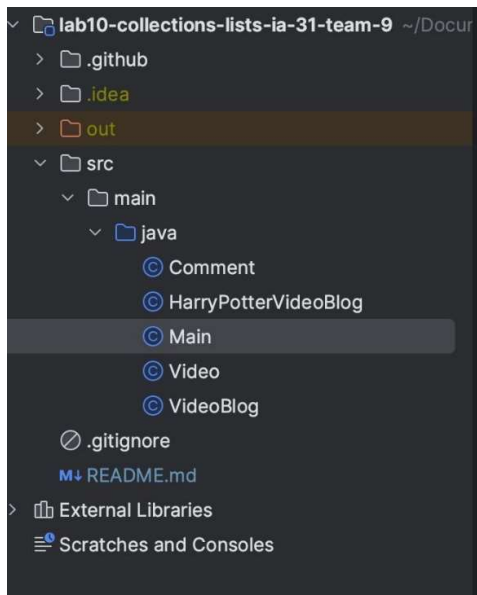


Рис. 5 структура лаб. 10 1 семестру

```

Gets the total watches of all the videos in the list of videos stored in VideoBlogs and sums them.
Demonstrates the usage of the typed Iterator interface
Params: videoBlog - the parameter of the function.
Returns: sum of total watches of video.

1 usage: java cyber...fire
private static int getVideoTotalWatch(VideoBlog videoBlog) {
    int result = 0;

    Iterator<Video> videoIterator = videoBlog.getVideos().iterator();

    while (videoIterator.hasNext()) {
        Video video = videoIterator.next();
        result += video.getWatchCount();
    }
    return result;
}

```

Рис. 6 Метод 1 лаб. 10 1 семестру

```

Checks if there is a comment that has more likes than videos.
Demonstrates the usage of the un-typed Iterator interface.
Params: videoBlog - the parameter of the function.
Returns: true - if there is a comment with more likes than the video.

1 usage  ▲ cyber_fire
private static boolean isCommentLikesGreaterThenVideoLikes(VideoBlog videoBlog) {
    Iterator videoIterator = videoBlog.getVideos().iterator();

    while (videoIterator.hasNext()) {
        Video video = (Video) videoIterator.next();
        Iterator commentIterator = video.getComments().iterator();

        while (commentIterator.hasNext()) {
            Comment comment = (Comment) commentIterator.next();

            if (comment.getLikes() > video.getLikes()) {
                return true;
            }
        }
    }

    return false;
}

```

Рис. 7 Метод 2 лаб. 10 1 семестру

```

Compares the video and returns the one with more dislikes.
Demonstrates the usage of typed for-each loop.
Params: videoBlog - the parameter of the function.
Returns: list of the video with most dislikes

1 usage  ▲ cyber_fire
private static List<Video> getVideosWithMostDislikes(VideoBlog videoBlog) {
    int maxDislikes = Integer.MIN_VALUE;
    List<Video> videoList = new ArrayList<>();

    for (Video video : videoBlog.getVideos()) {
        if (video.getDislikes() > maxDislikes) {
            videoList.clear();
            videoList.add(video);
            maxDislikes = video.getDislikes();
        } else if (video.getDislikes() == maxDislikes) {
            videoList.add(video);
        }
    }

    return videoList;
}
}

```

Рис. 8 Метод 3 лаб. 10 1 семестру

```

<jsp:useBean id="videoList" scope="request" type="java.util.ArrayList"/>
<jsp:useBean id="maxDislikes" scope="request" type="java.lang.Integer"/>
<jsp:useBean id="videoBlog" scope="request" type="com.harrypottervideoblog.model.VideoBlog"/>

<c:forEach items="${videoBlog.videos}" var="video">
    <c:choose>
        <c:when test="${video.dislikes > maxDislikes}">
            <c:set var="noUse" value="{
                videoList.clear();
                videoList.add(video);
                maxDislikes = video.dislikes
            }"/>
        </c:when>
        <c:otherwise>
            <c:if test="${video.dislikes == maxDislikes}">
                <c:set var="noUse" value="{
                    videoList.add(video)
                }"/>
            </c:if>
        </c:otherwise>
    </c:choose>
</c:forEach>

<h1>Videos With Most Dislikes:</h1>

<c:forEach items="${videoList}" var="video">
    <h1><c:out value="${video}"/></h1>
</c:forEach>

<button style="..." onclick="history.back()">Back</button>

</body>
</html>

```

Рис. 9 Відповідна логіка в лаб. 4 до методу 3

```

1 <% taglib uri="jakarta.tags.core" prefix="c" %>
2 <% page contentType="text/html; charset=UTF-8" %>
3 <html>
4 <head>
5 <title>Harry Potter Video Blog</title>
6 </head>
7 <body>
8
9 <c:set var="commentLikesGreaterThanVideoLikes" value="false" />
10 <jsp:useBean id="videoBlog" scope="request" type="com.harrypottervideoblog.model.VideoBlog"/>
11
12 <c:forEach items="${videoBlog.videos}" var="video">
13 <c:forEach items="${video.comments}" var="comment">
14 <c:if test="${comment.likes > video.likes && !commentLikesGreaterThanVideoLikes}">
15 <c:set var="commentLikesGreaterThanVideoLikes" value="true"/>
16 </c:if>
17 </c:forEach>
18 </c:forEach>
19
20 <h1>Is Comment Likes Greater Than Video Likes: ${commentLikesGreaterThanVideoLikes}</h1>
21
22 <button style="..." onClick="history.back()">Back</button>
23 </body>
24 </html>

```

Рис. 10 Відповідна логіка в лаб. 4 до методу 2

```

1 <% taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2 <% page contentType="text/html; charset=UTF-8" %>
3
4 <html>
5 <head>
6 <title>Video Total Watch</title>
7 </head>
8 <body>
9
10 <c:set var="result" value="0" property="number"/>
11 <jsp:useBean id="videoBlog" scope="request" type="com.harrypottervideoblog.model.VideoBlog"/>
12
13 <c:forEach items="${videoBlog.videos}" var="video">
14 <c:set var="result" value="${result + video.watchCount}" property="number"/>
15 </c:forEach>
16
17 <h1>Video Total Watch : ${result}</h1>
18
19 <button style="..." onClick="history.back()">Back</button>
20
21 </body>
22 </html>

```

Рис. 11 Відповідна логіка в лаб. 4 до методу 1

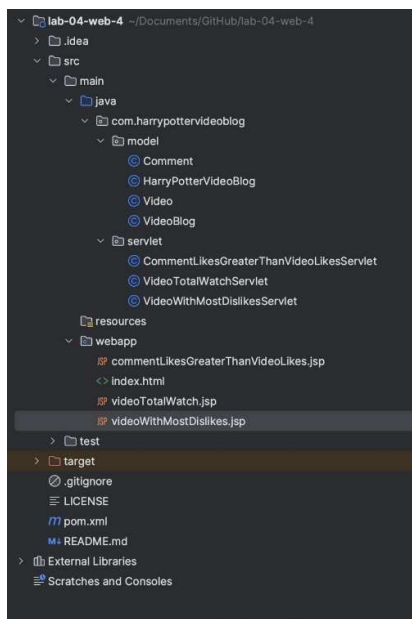


Рис. 12 структура проекту лаб. 4

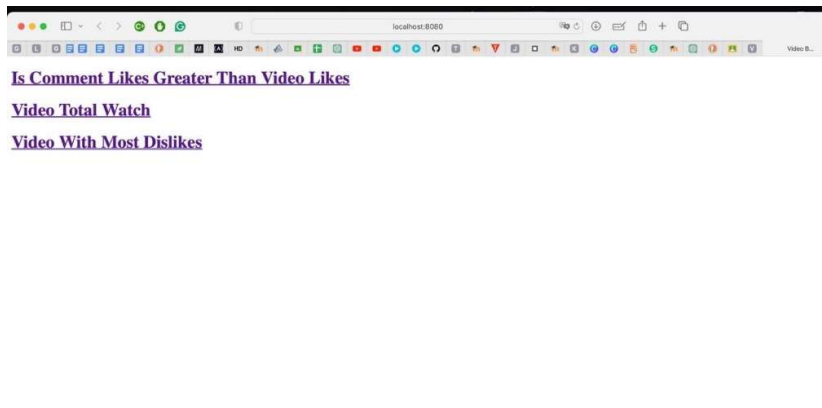
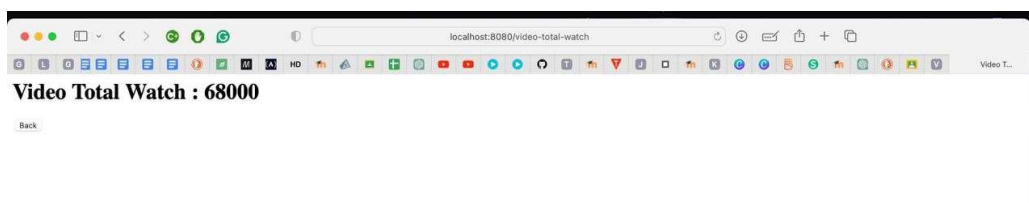


Рис. 13 головна стрінка – навігатор



**Рис. 14 відображення чи є в нас хочаб 1 відео, де лайки його коментарів
бальші за лайки самого відео (метод 2 з лаби 10)**



**Рис. 15 відображення загальної кількості переглядів всіх відео в блозі
(метод 1 з лаби 10)**



**Рис. 16 Відображення відео в списку які отримали найбільшу кількість
дізлайків (в нашому випадку лише 1) (логіка метод 3 з лаби 10)**

Висновок:

Використання MVC шаблону допомагає розділити модель (Model), клас який відповідає за web (Controller) та дані які треба відобразити в JSP (View).

Також використання EL і JSTL було дуже корисним для уникнення використання коду Java в View. Якщо в нас не має доступу в JSP до коду Java, наш додаток набагато більш забезпечений (security) від, наприклад, Security Leaks.