

**Міністерство освіти і науки України
Національний аерокосмічний університет
ім. М.Є. Жуковського**

Кафедра 503

Лабораторна робота № 6

З дисципліни «Кросплатформенні технології»

**Тема: «Ознайомлення з технологією створення і використання сервлетів на
мові програмування Java»**

Виконав:

студент групи 535 ст 1 Гужва М.А.


Харків 2020

Мета роботи:

1. Ознайомлення з технологією створення і використання сервлетів на мові програмування Java.
2. Ознайомлення з JDBC (Java DataBase Connectivity) API. Засвоєння класів Connection, DatabaseMetaData, Statement, ResultSet пакета java.sql.

Хід виконання роботи

1. Сучасні кросплатформені технології Java підтримують використання сервлетів, спеціальних класів, що обробляють запити до серверу в контексті веб-застосунків. Для створення сервлету, необхідно створити maven проект, або інший варіант веб-дадатку Java, далі завантажити Tomcat або Jetty сервер та надати шлях до проекту до серверу. У папці src/main створюються виконувані сервлет-класи, що мають import javax.servlet.* та обробляють doGet() метод.



The screenshot shows an IDE with the following components:

- EXPLORER:** Displays the project structure for 'SIMPLE-SERVLET-MASTER'. It includes folders for 'main', 'java', 'webapp', and 'WEB-INF'. The 'Counter.java' file is highlighted under the 'java' folder.
- OUTLINE:** Shows the 'JETTY SERVERS' section with 'jetty-distributio...' and 'simple-servlet-...' listed.
- Code Editor:** Displays the code for 'Counter.java'. The code is as follows:

```
1 import java.io.*;
2 import javax.servlet.*;
3 import javax.servlet.http.*;
4
5 public class Counter extends HttpServlet {
6     int accesses = 0;
7
8     public void doGet(HttpServletRequest request, HttpServletResponse response)
9         throws IOException, ServletException
10    {
11        response.setContentType("text/html");
12        PrintWriter out = response.getWriter();
13
14        accesses++;
15        out.print("Number of times this servlet has been accessed:" + accesses);
16    }
17 }
```

Рисунок 1 – Приклад роботи простого сервлету Java

Результат роботи простого розробленого сервлету – вивід інформації до html-сторінки, див. рис. 2. Діаграма потоків даних наведена на рис. 3.

Directory: /simple-servlet-0.1/

Рисунок 2 – Вивід строки до html

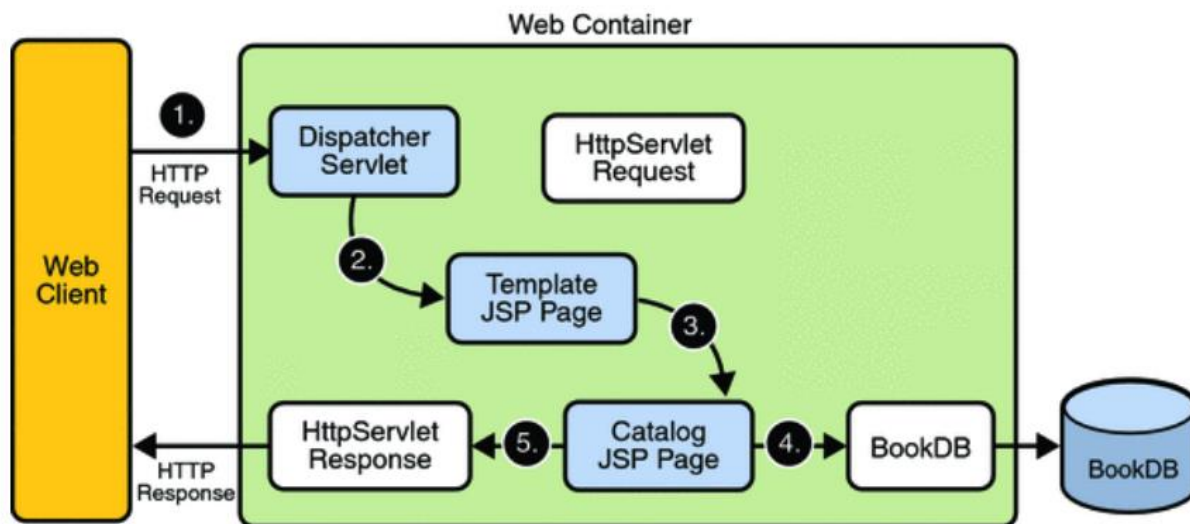


Рисунок 3 – Діаграма потоків даних до розроблюваного прикладу

Таким чином, в ході розробки простого сервлету, було розглянуто принцип організації та роботи веб-орієнтованих додатків на Java.

2. Ознайомлення з JDBC (Java DataBase Connectivity) API. Засвоєння класів Connection, DatabaseMetaData, Statement, ResultSet пакета java.sql.

Продовженням ознайомлення з принципами роботи JDBC є створення анонімайзера для даних таблиць БД, на основі знань попередньої лабораторної роботи. Використовуючи створений раніше клас, додамо строки бази даних, що відповідають користувачам магазину. За завданням необхідно додати 100 записів. Кожен запис створимо випадково, з набору з 12 імен та прізвищ. Вік людини задається випадковим чином рис.3.

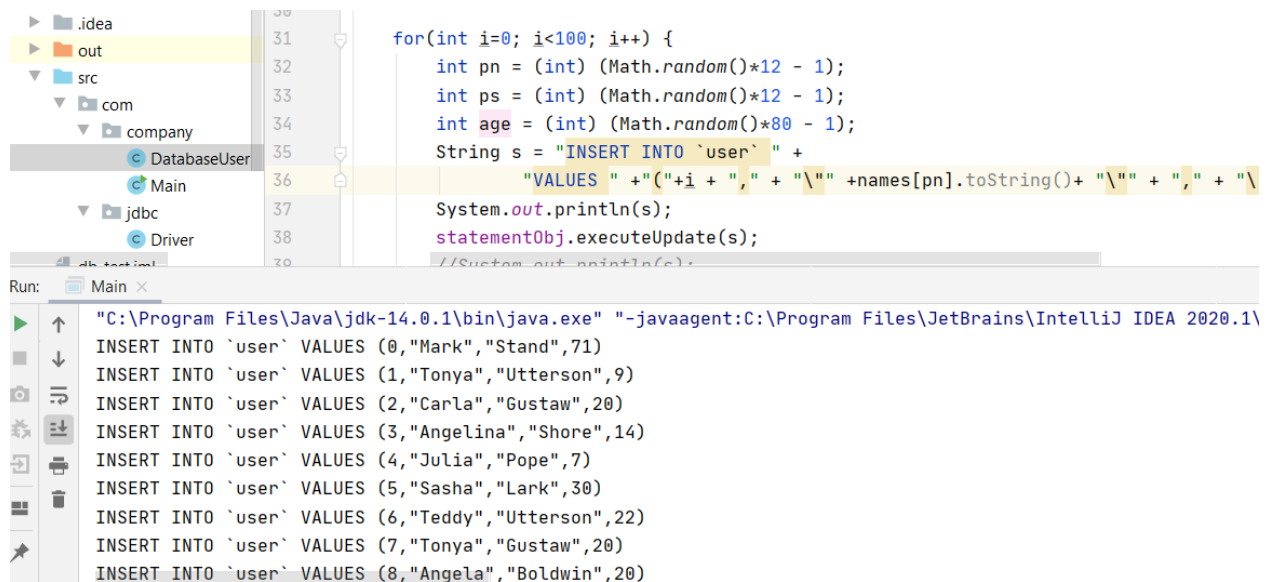


Рисунок 3 – Генерація строк запитів до бази даних зі згенерованими додатком записами користувачів

Паралельно, записи користувачів додаються у внутрішній масив додатку, для подальшої анонімізації.

Структура БД записів користувачів на рис.4.

The screenshot shows the phpMyAdmin interface. The left sidebar displays a database structure tree. The main area shows the structure of the 'user' table in the 'store' database.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(50)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 3	surname	varchar(50)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 4	age	int(11)			Нет	Нет		

Рисунок 4 – Структура записів користувачів

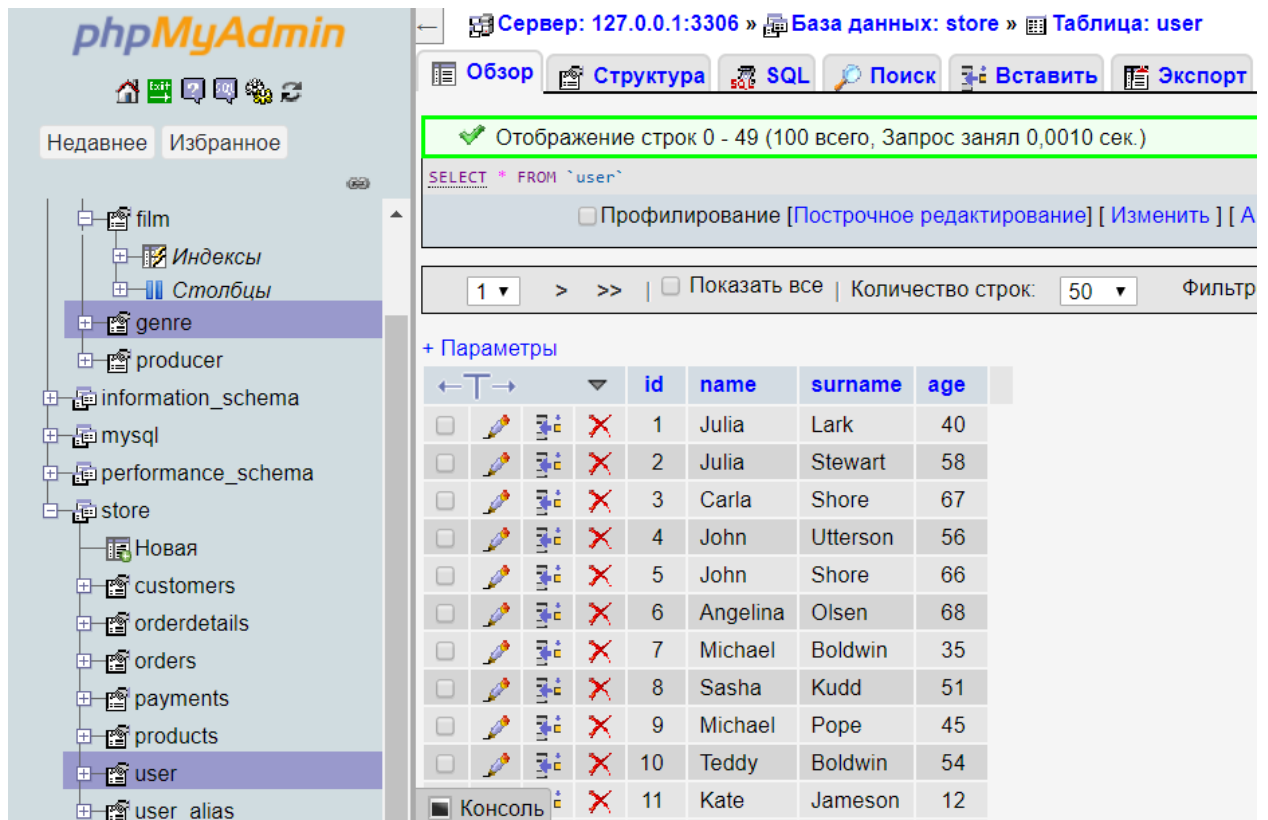


Рисунок 5 – Додані програмою дані користувачів

На рис. 6 продемонстровано створення аліасів(псевдонімів) записів до бази даних, дані також окремо зберігаються в масиві програми String[] changedNames, та пишуться до окремої таблиці БД user_alias.

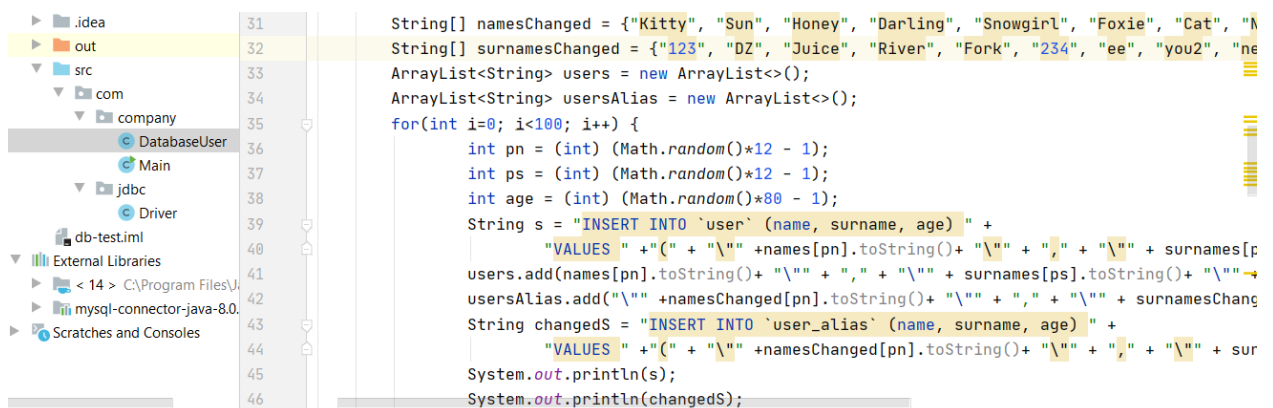


Рисунок 6 – Паралельне створення дійсних даних та псевдонімів у циклі на 100 записів

На рис. 7 зображено вивід у консоль інформації про запис даних до БД.

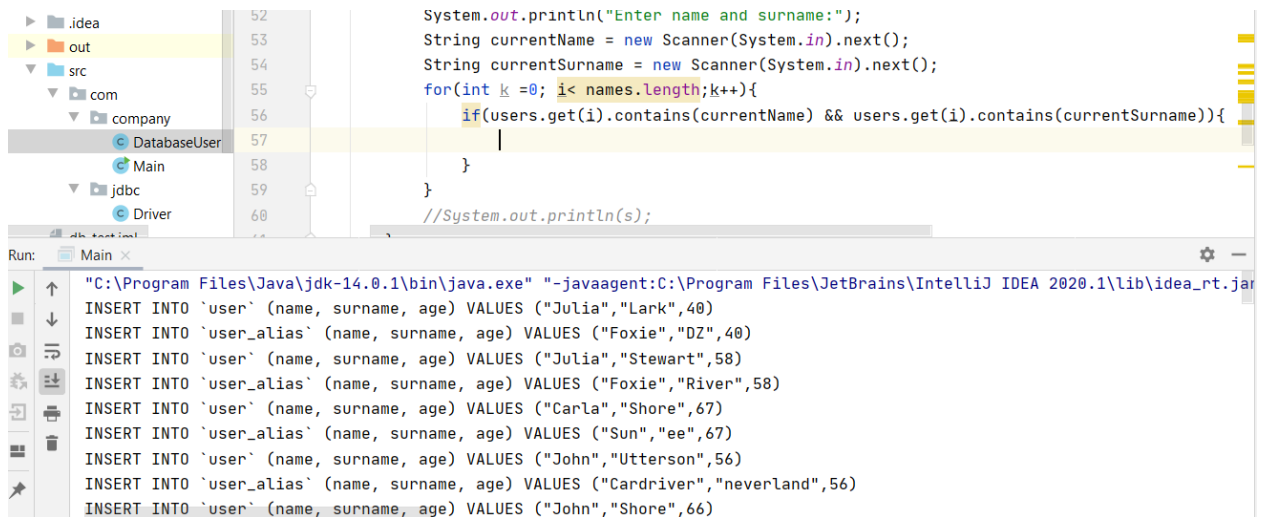


Рисунок 7 – Паралельний запис до 2-х таблиць БД даних користувачів

Тепер можна організувати пошук по імені та прізвищу даних таблиць БД, на основі `ArrayList<String>` `user`, `user_alias`, куди паралельно записувалися дані БД. Організація подібного пошуку наведена нижче:

```

System.out.println("Enter name and surname:");
String currentName = new Scanner(System.in).next();
String currentSurname = new Scanner(System.in).next();
for(int k = 0; k < users.size(); k++){
    if(users.get(k).contains(currentName) && users.get(k).contains(currentSurname)){
        System.out.println("Your persons alias is");
        System.out.println(usersAlias.get(k));
    }
}

```

Приклад відшукування записів наведено на рис. 8.

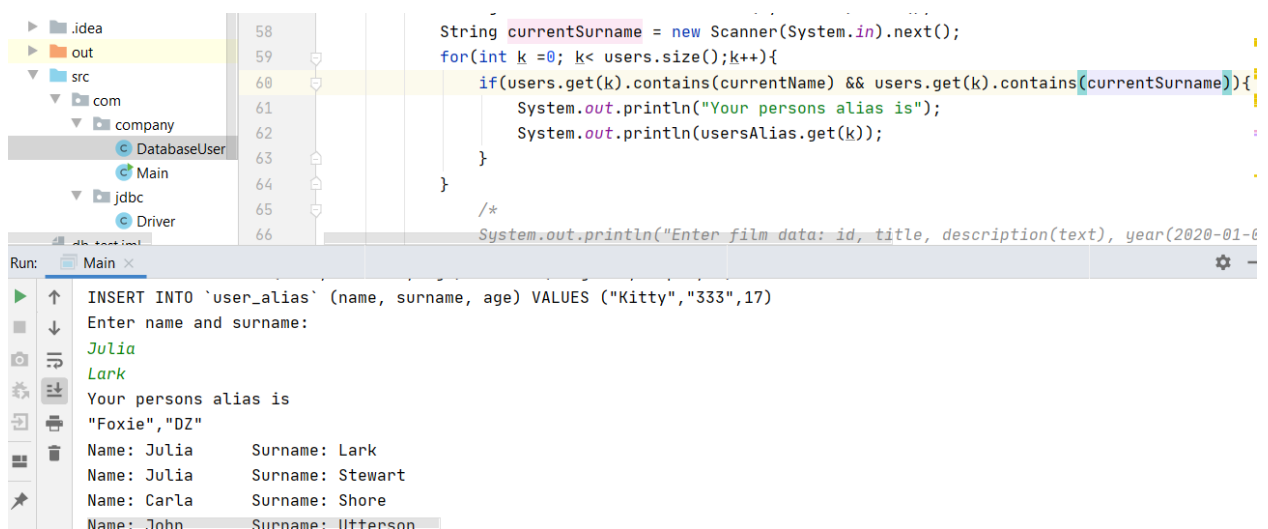


Рисунок 8 – Відшукування даних псевдонімів користувачів

Таким чином, детально було розглянуте питання роботи з базами даних, в контексті роботи з анонімними даними та дійсними даними користувачів.

Висновки: в ході виконання лабораторної роботи були розглянуті питання створення та налагодження веб-орієнтованих додатків з використанням технології сервлетів на Java а також закріплення навичок роботи з базами даних на прикладі створення анонімних даних користувачів. Добуті вміння допоможуть в подальшому розробляти складні веб-орієнтовані додатки, використовуючи технології сервлетів, JDBC та баз даних, налаштовувати сервери Tomcat та Jetty та підтримувати вже розроблені додатки.