Національний університет “Одеська політехніка”

Кафедра комп’ютеризовані системи та програмні технології

**КУРСОВА РОБОТА**

з дисципліни «Сучасні інтернет-технології»

Варіант 5

Студента 1 курсу, групи АТ–212

спеціальності «Автоматизація та комп'ютерно-інтегровані технології»

Лефтера О. Г.

Керівник: доц. Сперанський В.О.

Національна шкала:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

     Члени комісії  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   \_\_\_\_\_Сперанський В.О.\_\_\_\_\_

(підпис)          (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_Ілуца А.С.\_\_\_\_\_\_\_\_\_\_\_

(підпис)          (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)          (прізвище та ініціали)

м. Одеса – 2022 рік

1. Спроектувати структуру інформаційної бази свого варіанта завдання (Додаток А).
2. Створити основну таблицю інформаційної бази як динамічного списку об'єктів.
3. Передбачити функції контролю коректності всіх даних, що вводяться (наприклад, невід'ємний вік, дата продажу не раніше, ніж 100 років тому тощо).
4. Заповнити основну таблицю щонайменше 20 записами.
5. Забезпечити збереження та завантаження даних у зовнішній файл/з файлу.
6. Реалізувати інтерфейс користувача, який забезпечуватиме наступні функції:
7. створення інформаційної бази тієї предметної галузі, яка визначена постановкою завдання;
8. додавання нових записів до бази даних (з контролем коректності значень, що вводяться) в окремому вікні (модальному);
9. пошук записів по ключовому полю;
10. редагування записів бази даних у окремому вікні (модальному);
11. видалення непотрібних записів із підтвердженням дії;
12. збереження даних у файлі;
13. видачу звітів за запитом із можливістю сортування;
14. реалізувати систему нотифікацій в інтерфейсі, яка повідомлятиме про успішність/провальність дій в інтерфейсі (додавання запису, видалення тощо
15. Виконати оригінальну обробку, вказану у постановці конкретного завдання для вашого варіанту (Додаток А).
16. Організувати введення пароля під час операцій змін даних.
17. Додати довідкову систему користувача за створеною програмою.
18. Підготувати пояснювальну записку за розробленим проектом.

**ДОДАТОК (А)**

Сформувати базу даних, куди заносять результати змагань зимової Олімпіади. Кожен запис містить:

* реєстраційний номер спортсмена;
* ПІБ спортсмена;
* назву країни (Україна, США тощо);
* вид змагань (бобслей, ковзани та ін.);
* показаний результат;
* зайняте місце.

Передбачити:

1. видачу списку призерів з кожного виду спорту;
2. видачу довідок про кількість золотих, срібних та бронзових медалей, завойованих кожною країною - учасницею Олімпіади;
3. видачу відомостей про всіх призерів країни із заданою назвою;
4. видалення з бази записів про спортсменів із зазначеним прізвищем (на вимогу допінг-контролю);

**ЕТАПИ ПРОЕКТУВАННЯ**

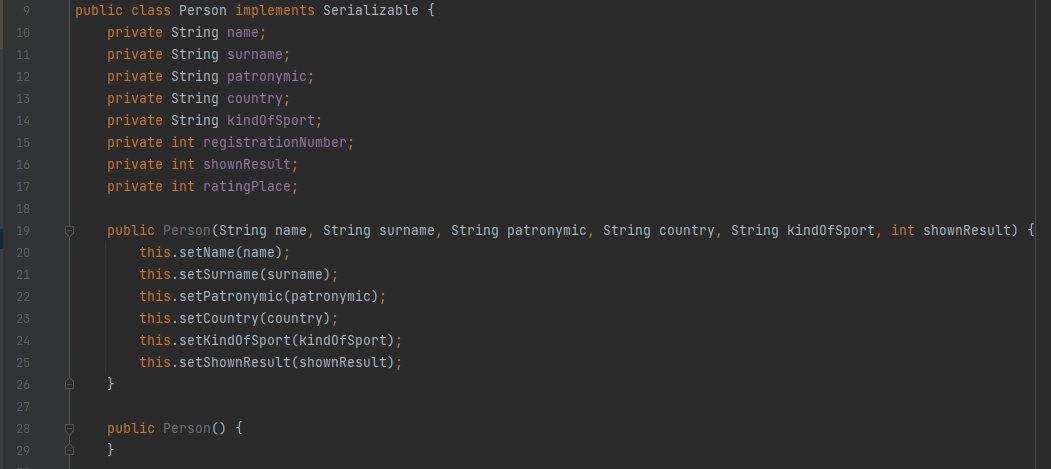
|  |  |  |  |
| --- | --- | --- | --- |
| № | Найменування | Зміст | Дата закінчення |
| 1. | Початок | Вивчення постановочного завдання, предметної галузі | 10.05.2022 |
| 2. | Теоретична підготовка | Вивчення літератури з дисципліни | 15.05.2022 |
| 3. | Підготовка базових класів | Створення класів для додавання, видалення в список, та генерації об’єктів | 20.05.2022 |
| 4. | Створення  первинної  бази даних | Створення класів для серіалізації/десеріалізації, та заповнення файлу | 25.05.2022 |
| 5. | Верстка користувацького інтерфейсу | Створення інтерфейсу для  виводу даних, сортування та створення модальних вікно для редагування списку об’єктів. Адаптація для різних екранів | 10.06.2022 |
| 6. | Запуск сервера | Створення та налаштування  класів для сервера. Створення  функцій для запитів до сервера | 12.06.2022 |
| 7. | Реалізація зв’язку користувацького інтерфейсу з сервером | Ввід та вивід даних, видача  довідок, створення перевірок  даних, вводу пароля | 12.06.2022 |

Використані технології:

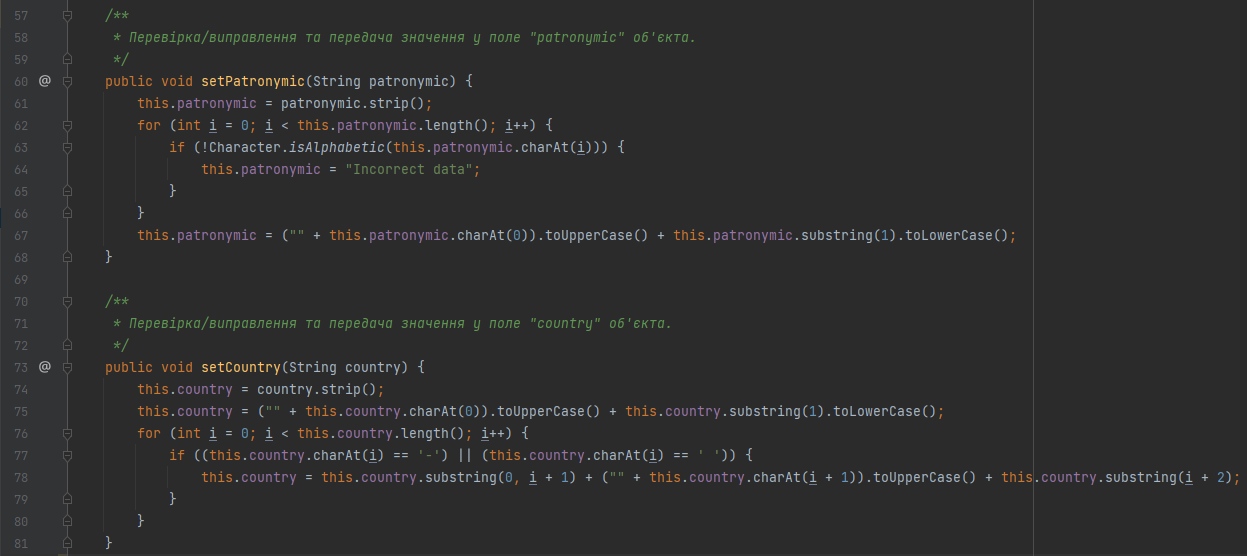
Java, fasterxml.jackson, HttpServer, HTML, CSS, JS

**ДЕТАЛЬНИЙ ОПИС ПРОЕКТУ**

**BACK-END**

1. Створення класу «Person», для зберігання даних про кожного спортсмена-учасника зимової Олімпіади, створюються поля та конструктор(рис. 1.1).

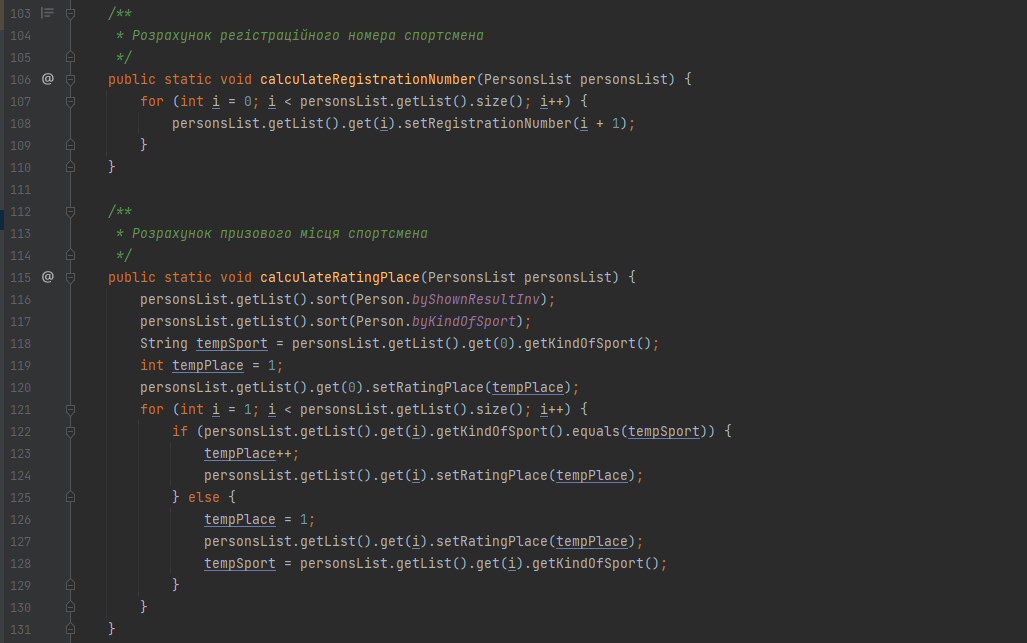
(рис. 1.1)

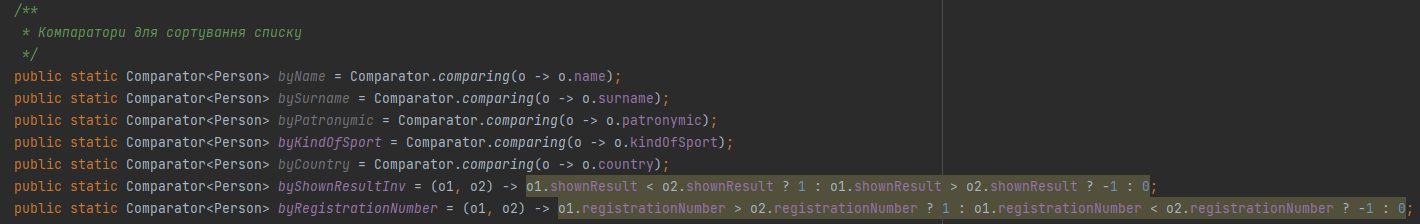
Також створення setter’s та getter’s, при цьому, спочатку значення потрапляють із конструктора в setter, за для перевірки та приведення даних у правильний вигляд(рис. 1.2, 1.3, 1.4).

(рис. 1.2)

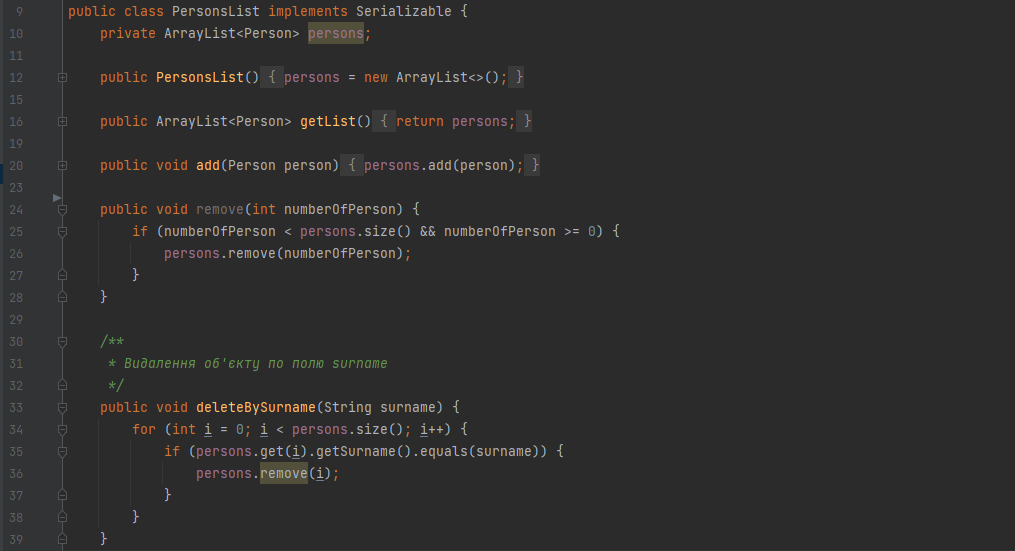
(рис. 1.3)

(рис. 1.4, в наступних setter’s немає перевірок через те, що дані поступають одразу перевірені відредаговані)

Додані функції, при виклику яких підраховується реєстраційний номер та рейтингове місце учасника. Реєстраційний номер присвоюється із положення об’єкту в списку, тобто при додаванні буде присвоюватися номер на 1 більший від номеру останнього учасника. Місце учасника вираховується автоматично, враховуючи набрані бали та окремий рейтинг для кожного виду спорту. (див. рис. 1.5)

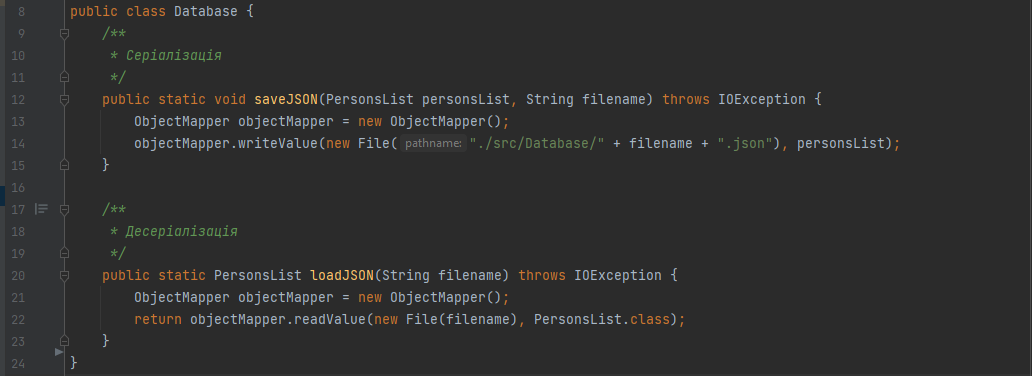
(рис. 1.5, показані функції потрібно викликати щоразу, як змінюються дані)

(рис. 1.6, Компаратори для сортування об’єктів)

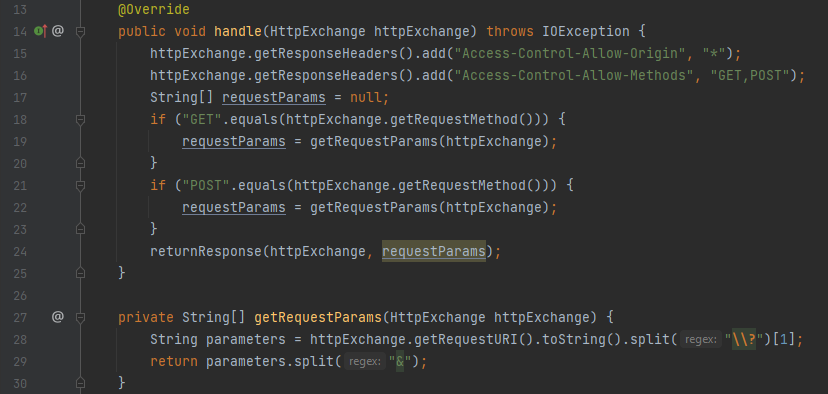
2. Створення класу «PersonsList» зі списком, для всіх об’єктів в одному списку, для сортування і т. д. Створення методів для додавання об’єктів у список та видалення з нього об’єктів за вказаним прізвиськом. (див. рис. 1.7)

(рис. 1.7)

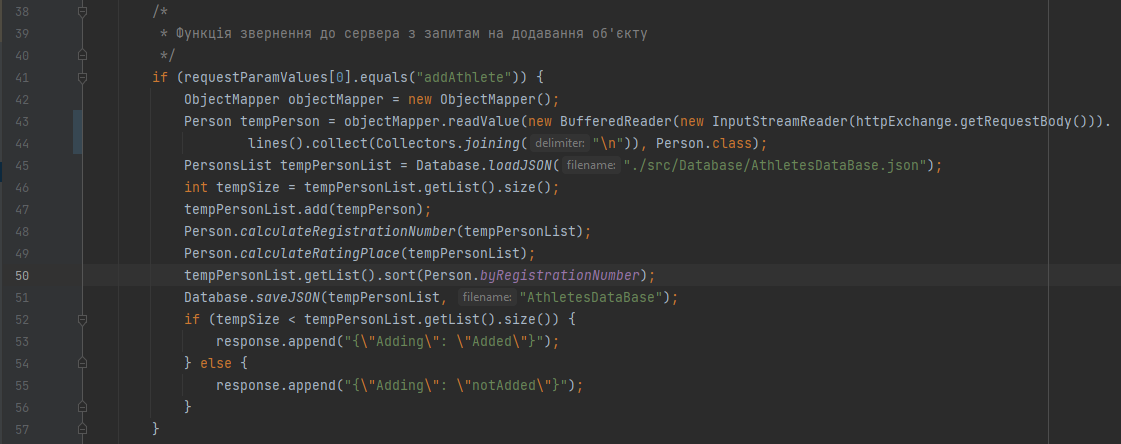
3. Створення класу «Database» з методами серіалізації та десеріалізації списку об’єктів.(передчасно, класи «Person» та «PersonsList» реалізують інтерфейс «java.io.Serializable» для можливості серіалізації та десеріалізації) (див. рис. 8)

(рис. 1.8)

4. Створення класу «ServerOlympic», який керує працею сервера, отриманням та відправкою даних, та методів та функцій, які надають можливість це робити.

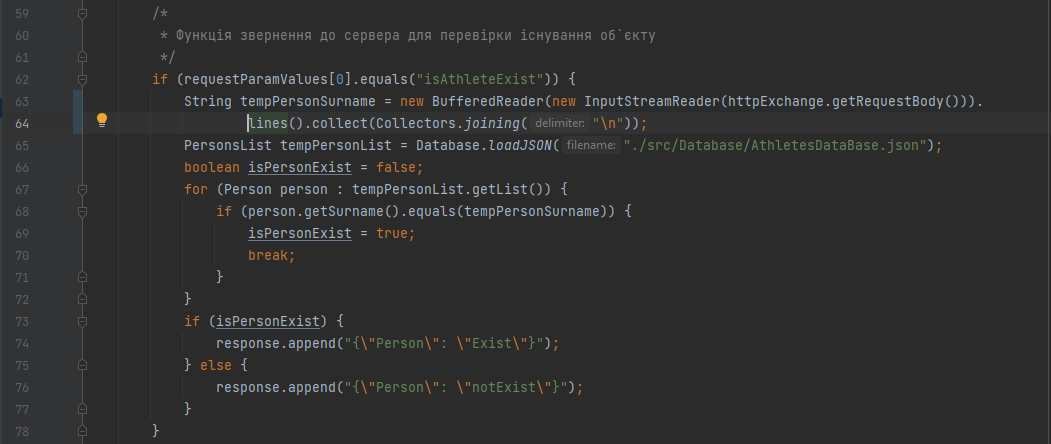
 (рис. 1.9, отримання запитів та передача їх далі)

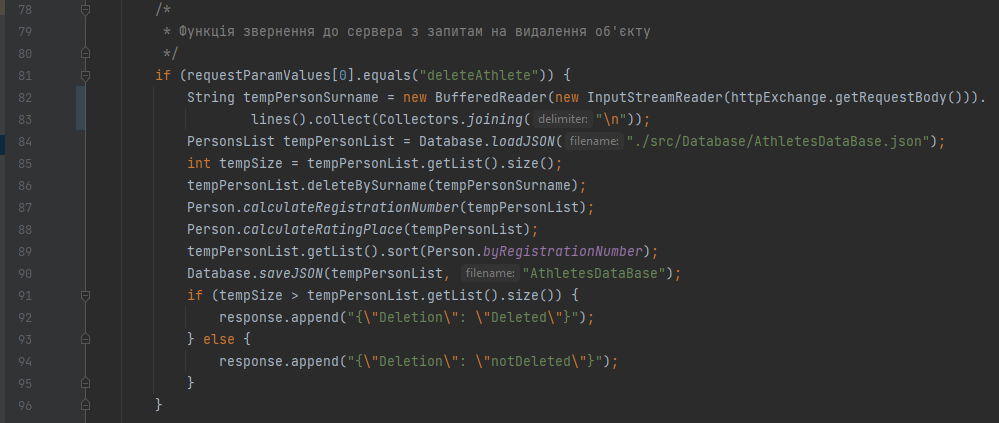
Обробка запита «addAthlete» на додавання нового спортсмена у список. Отримання даних з клієнтської бази(JSON рядок який містить поля та їх значення, об’єкта Person), дешифрування та зберігання його у тимчасову змінну, десеріалізація основного файлу бази даних у тимчасовий список об’єктів, та додавання методом .add() об’єкту в список, виклик функцій перерахунку рейтингу та реєстраційного номеру, та подальша серіалізація оновленого списку в файл. Також передбачена перевірка успішності операції, за допомогою порівняння розміру списку до додавання та після, якщо нова довжина списку більша за стару, то операція вважається успішною та сервер надсилає дані про успішність. (див. рис. 1.10)

(рис. 1.10)

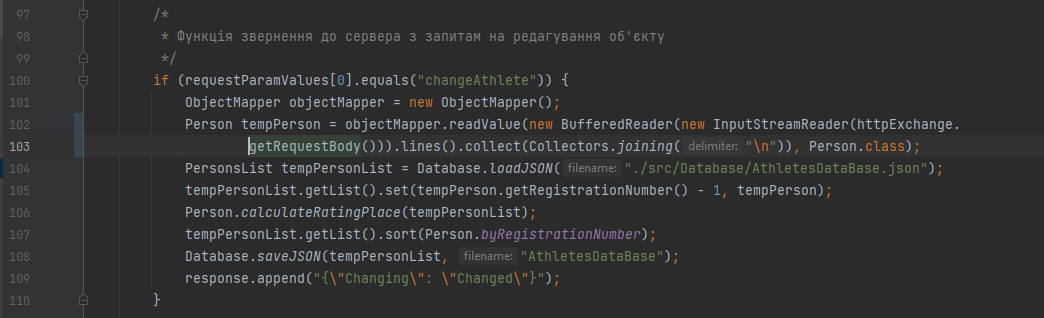
Обробка запиту «isAthleteExist» на наявність спортсмена з переданим прізвиськом, та подальша відправка відповіді від сервера про наявність або відсутність(потрібно для видалення спортсмену зі списку).

Сервером по запиту отримується текстове значення прізвиська, база даних десеріалізується у тимчасовий список, та методом .equals() перевіряється наявність хоча би одного спортсмена з таким прізвищем. (див. рис. 1.11)

(рис. 1.11)

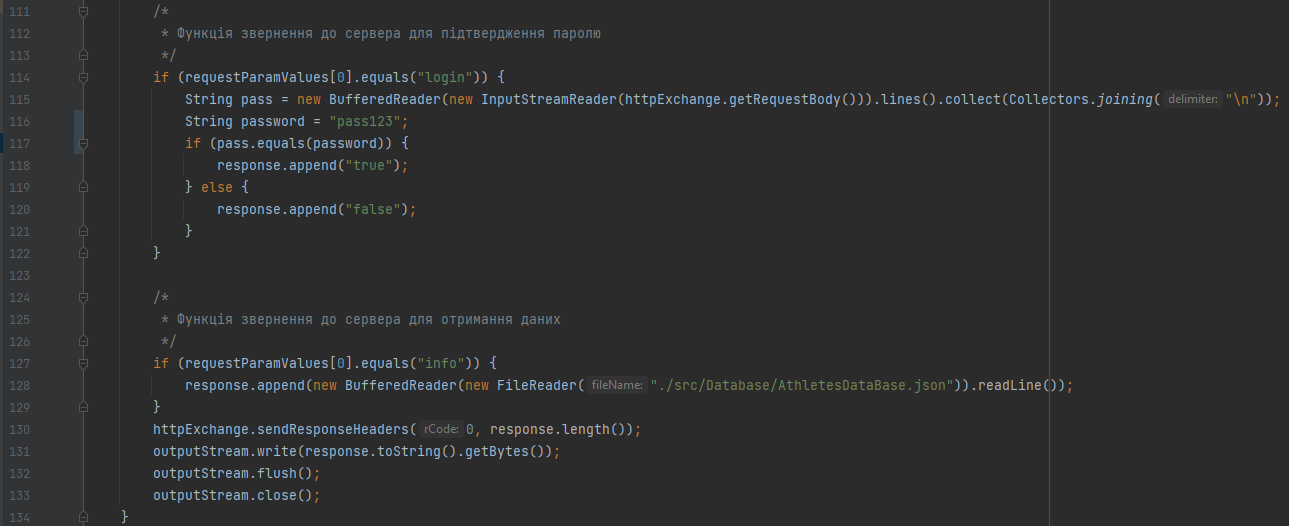
 Обробка запиту на видалення спортсмена по заданому прізвищу. Сервер отримує строкове значення прізвища, десеріалізує базу даних у тимчасовий список, методом .deleteBySurname() видаляє елемент із списку, викликаються функції для перерахунку місць та номерів, список десеріалізується назад та відправляється відповідь про виконання операції. (див. рис. 1.12)

(рис. 1.12)

 Обробка запиту «changeAthlete» на змінення будь яких даних про вибраного спортсмена. Сервер отримує JSON-строку з об’єктом та записує у тимчасовий об’єкт, десеріалізує файл-базу даних у тимчасовий список, та методом .set(), об’єкт з тим самим значенням реєстраційного номера переприсвоюється на новий тимчасовий об’єкт, перераховуються рейтингові місця і список серіалізується назад у файл. Сервер відправляє відповідь про успішність операції. (див. рис. 1.13)

(рис. 1.13)

Обробка запиту підтвердження пароля «login» потрібна для підтвердження дій зміни бази даних. Сервер отримує строкове значення та порівнює його із заданим паролем. При співпадінні, сервер надсилає відповідь про співпадіння. (див. рис. 1.14)

 Обробка запиту на отримання інформації «info» зчитує дані із файлу .json та відправляє строку. (див. рис. 1.14)

(рис. 1.14)

У класі «Main» створений метод для запуску сервера на <http://localhost:8080/back>, та це метод запускається та запускає сервер.

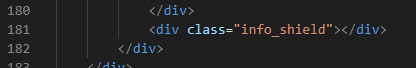
(рис. 1.15)

**FRONT-END**

1. Верстка клієнтського інтерфейсу для відображення, змінення, сортування, пошуку інформації, створення модальних вікон та зв’язок їх між собою, та стилізування за допомогою CSS, та адаптація під мобільні пристрої.

(рис. 2.1, вікно довідки про медалі та вікно анкети для додавання спортсмена,(вікно редагування таке саме))

(рис. 2.2, вікно видалення спортсмена, вікно підтвердження дії та вікно вводу пароля відповідно послідовності)



(рис. 2.3, блок у який додаються банери з інформацією про кожного спортсмена)

(рис. 2.4, блок сортування)

2. Сортування та видача довідок:

(рис. 2.5, відповідно відміна сортування, живий пошук)

(рис. 2.6, видача довідок про кількість медалей у кожної країни)

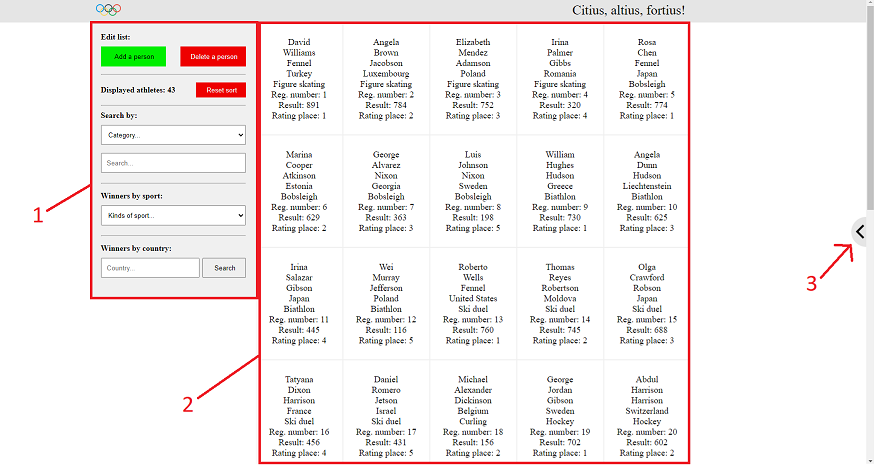
(рис. 2.7, генерація банерів з інформацією про кожного спортсмена)

(рис. 2.8, функції додавання та видалення об’єктів)

(рис. 2.9, функція підтвердження пароля)

**КОРИСТУВАЦЬКА ІНСТРУКЦІЯ**

ГОЛОВНА СТОРІНКА

Загальний вигляд головної сторінки (див. рис. 3.1)

(рис. 3.1)

На головній сторінці можна побачити кілька ключових елементів(див. рис. 3.1):

1. Вікно для пошуку/сортування списку учасників зимової Олімпіади, та редагування списку учасників.
2. Вікно виводу інформації про учасників.
3. Кнопка для відкриття інформаційної панелі з довідкою про кількість медалей, набраних кожною країною-учасницею.

ПОШУК ТА ВИДАЧА ДОВІДОК



(рис. 3.2)

1) Стрілкою 1 показане поле, у яке виводиться кількість показаних спортсменів. (див. рис. 3.2)

2) Для пошуку наживо, у списку, який позначено вказівником 2, виберіть категорію по якій треба виповнити пошук, та у полі 3, введіть дані. (див. рис. 3.2)

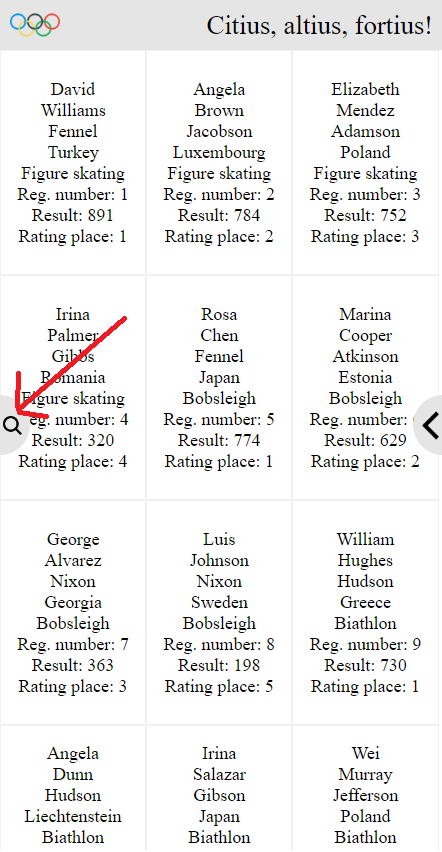
3) Для отримання списку призерів по кожному виду спорту, виберіть вид спорту у списку, який позначено вказівником 4. (див. рис. 3.2)

4) Для пошуку призерів по країні, впишіть її назву у поле 5, та натисніть кнопку «Search» (див. рис. 3.2)

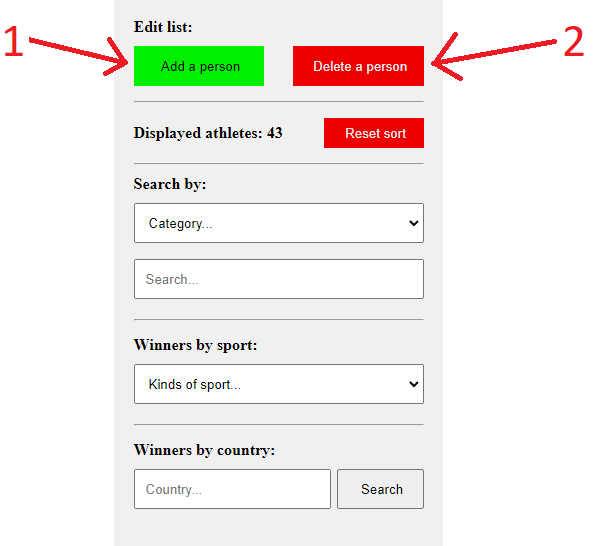
5) Для скидання пошуку, натисніть кнопку під номером 6. (див. рис. 3.2)

6) Для отримання довідки про кількість золотих, срібних та бронзових медалей кожної країни, натисніть на кнопку зі стрілкою (позначеною цифрою 3) у правій частині екрану (див. рис. 3.1 та 3.3)

(рис. 3.3)

\*На мобільних пристроях, для відкриття вікна сортування, треба натиснути кнопку у лівій частині екрану(див. рис. 3.4)

(рис. 3.4)

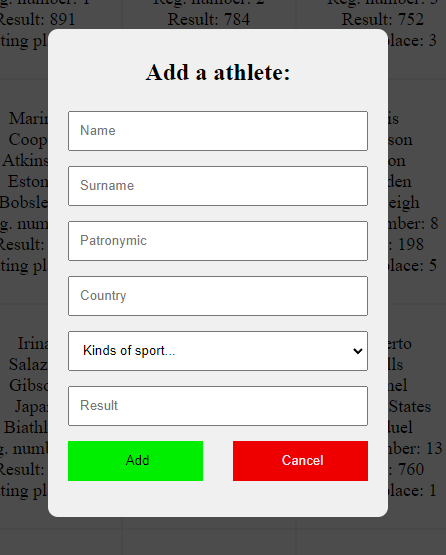
РЕДАГУВАННЯ СПИСКУ

(рис. 3.5)

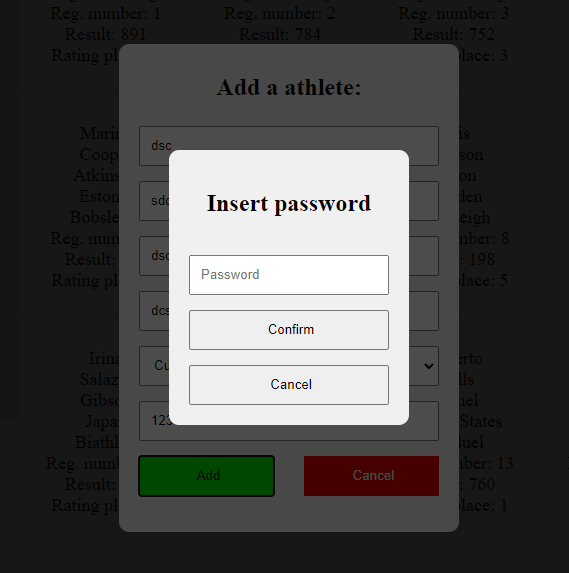
1) Для додавання нового спортсмена у список учасників олімпіади, потрібно натиснути на кнопку, яку позначено стрілкою під номером 1.

Відкриється анкета (див. рис. 3.6), яку потрібно заповнити(латиницею), та натиснути кнопку «Add».

При натисканні «Add», з’явиться вікно для підтвердження паролю (див. рис. 3.7), введіть пароль\* та натисніть «Confirm».



(рис. 3.6)



(рис. 3.7)

2) Для видалення запису про спортсмена по прізвищу, натисніть кнопку 2 (див. рис. 3.5).

Відкриється вікно для вводу прізвища (див. рис. 3.8), введіть прізвисько потрібного спортсмена та натисніть «Delete».

При натисканні «Delete», з’явиться вікно для підтвердження паролю (див. рис. 3.7), введіть пароль\* та натисніть «Confirm».

3) Для змінення даних в запису певного спортсмену, на головному екрані (див. рис. 3.1), натисніть 2 рази ЛКМ на «банер» потрібного спортсмену (на мобільних пристроях потрібно зажати «банер» потрібного спортсмену).

З’явиться вікно для підтвердження паролю (див. рис. 3.7), введіть пароль\* та натисніть «Confirm».

(рис. 3.8)

Відкриється вікно редагування даних спортсмена (див. рис. 3.8), заповнене даними про нього, змінить потрібні данні про спортсмена та натисніть кнопку «Edit».

Підказка: будь яке спливаюче вікно можна закрити натиснув кнопку «Cancel»

\*Пароль для редагування списку спортсменів – pass123

**ВИСНОВОК**

При виконанні курсової роботи, була створена база даних та веб-додаток для доступу/перегляду/редагуванню даних. Для виконання поставленого завдання були використані сучасні інтернет технології та технології програмування (JAVA, HTML, CSS, JS)

Було досліджено процес та отримані навички створення локального сервера та його зв’язок із клієнтським інтерфейсом.

Дослідив та більш детально зрозумів сутність веб-додатків, інтернет технологій, серверів та побудови користувацького інтерфейсу.

**ДЖЕРЕЛА**

1. Jackson документація:

<https://www.mousedc.ru/learning/592-peredacha-post-dannykh-fetch-javascript/>;

2. HttpServer Java документація:

<https://docs.oracle.com/javase/8/docs/jre/api/net/httpserver/spec/com/sun/net/httpserver/HttpServer.html>;

3. Using the fetch API:

<https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch>;

4. Artifacts, IntelliJ IDEA:

<https://www.jetbrains.com/help/idea/working-with-artifacts.html#build_artifacts>;

5. JSON in JS:

<https://learn.javascript.ru/json>

6. Приклад HttpServer:

<https://github.com/it-school/SimpleServer.git>;

7. Java, коротко об Object, наслідуванні і інтерфейсах:

<https://youtu.be/99kHqzLvKYg>;