**Лабораторна робота №6**

з курсу «**Проектування інформаційних систем**»

на тему: «**Система автоматичного створення довідника користувача та оформлення**

**коду за допомогою Coding Convention»**

Виконав:

студент групи ДА-61

Попов Григорій

**Мета роботи:** за допомогою системи генерації довідника користувача створити документ у форматі PDF і HTML для архітектурної програмної моделі.

**Завдання:**

1. Для кожного з класів API в код програми додати теги з описом керівництва

користувача для архітектурної програмної моделі.

2. Обрати Coding Convention. Оформити код програми згідно Coding Convention.

2. Встановити налаштування системи автоматичного створення керівництва

користувача.

3. Згенерувати HTLM документацію керівництва користувача.

4. Згенерувати PDF документацію керівництва користувача.

**Оформлення результатів роботи**

Всі класи додатку було заанотовано використовуючи утиліту Javadoc. Детальніше саму документацію можна продивитись в межах даного репозиторію за наступним посиланням: [..\projects\Popov\Documents\JavaDoc\index.html](../projects/Popov/Documents/JavaDoc/index.html)

Також було використано утиліту SwaggerUI разом із спеціальними анотаціями для вихідного коду серверного додатку, щоб згенерувати документацію REST API. Її можна переглянути за наступним посиланням:

[..\projects\Popov\Documents\Swagger\SwaggerOAS.html](../projects/Popov/Documents/Swagger/SwaggerOAS.html)

Оскільки обидва набори документації є інтерактивними та можуть працювати в офлайн-режимі, немає потреби у генерації тієї ж самої документації в pdf-форматі, оскільки це не лише збільшить її розмір, а й сприятиме втрати частини її інтерактивності.

Код даного додатку було відформатовано згідно Java Code Convensions (<https://www.oracle.com/technetwork/java/codeconvtoc-136057.html>) з використанням спеціальних форматерів в рамках IDE “Eclipse”. Також при розробці додатку було використано JavaFX best practices щодо оформлення структури згідно із патерном MVP. Єдина відмінність, яка була здійснена та закріплена за проектом на рівні домовленності, є введення поннятя Component.

Component – це View, яке може бути перевикористано довільну кількість разів в межах батьківськіго View, *та яка* ***також*** *базується на FXML-шаблоні, дозволяючи при цьому заповнювати себе даними за допомогою відповідних методів*. Для досягнення цього було поєднано концепції контроллера та зображення, тож Component містить зручні методи для маніпулювання власним контентом і створюється/використовується іншим контроллером. В даному додатку створено лише один Component – UserTileComponent. Детальніше про нього можна дізнатися прогорнувши Javadoc-документацію.

На даному етапі код було суттєво почищено також від попереджень (warnings), тож задачу оформлення коду в рамках coding convension можна вважати виконаною.

**Висновок**

В рамках даної лабораторної роботи було вивчено можливості утиліти JavaDoc стосовно генерації документації користувача із вихідного коду. Також було порівняно декілька coding conventions для мови програмування Java і обрано Java Code Convensions (від Oracle), оскільки вона простіша за аналогічну від Google тож більш доречна для такого маленького проекту як розробляємий додаток.