

Міністерство освіти і науки України
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Навчально-науковий комплекс «Інститут прикладного системного аналізу»
Кафедра системного проектування

Лабораторна робота №7

з дисципліни

“Проектування інформаційних систем”

Система автоматичного створення довідника користувача та оформлення
коду за допомогою Coding Convention

Виконала:

студентка групи ДА-72

Мелкозерова Ольга

Київ – 2020

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

Задача:

1. Вивчити теги системи генерації керівництва користувача.
2. Створити опис для всіх класів API з описом призначення кожного класу, методів класу і членів класу.
3. Згенерувати документацію у форматах PDF, HTML.

Хід роботи

З Coding Convention для Java було ознайомлено за посиланням: <https://www.oracle.com/java/technologies/javase/codeconventions-contents.html>

Для виконання даної лабораторної роботи використаємо Javadoc. Javadoc це генератор документації в HTML форматі з коментарів вихідного коду на java, він також є стандартом документування.

Він вбудований в IntelliJ IDEA, де були виконані мої попередні лаби, тому ніяких додаткових підключень не потребується.

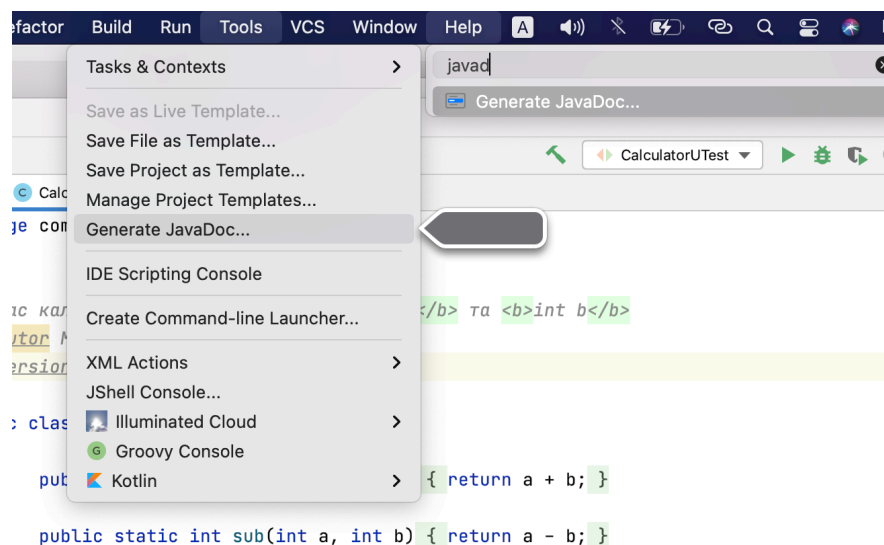


Рис.1 Стандартний засіб для генерації Javadoc

Документування класу, методу чи змінної починається з комбінації символів `/**`, після якого слідує тіло коментарів, а закінчується все `*/`. В тіло коментарів можна вставляти разін дескрптори, що починаються з `@` та нової строки.

Так, наприклад, повертаючись до програми калькулятора з Unit тестами з лабораторної роботи №5, напишемо базову документацію:

```
/**
 * Клас калькулятору з параметрами int a та int b
 * @author Мелкозерова Ольга ДА-72
 * @version 1.1
 */
public class Calculator { ... }
```

Змінні виділимо HTML-тегом `bold` для того щоб вони були жирні при генерації.

Також задокументуємо кожен з методів (напр. `Add`):

```
/** Функція додавання
 * @param a - перший доданок
 * @param b - другий доданок
 * @return повертає суму a та b*/
public static int add(int a, int b){
    return a + b;
}
```

Перейдемо до класу тестів, за тим же принципом документуємо його.

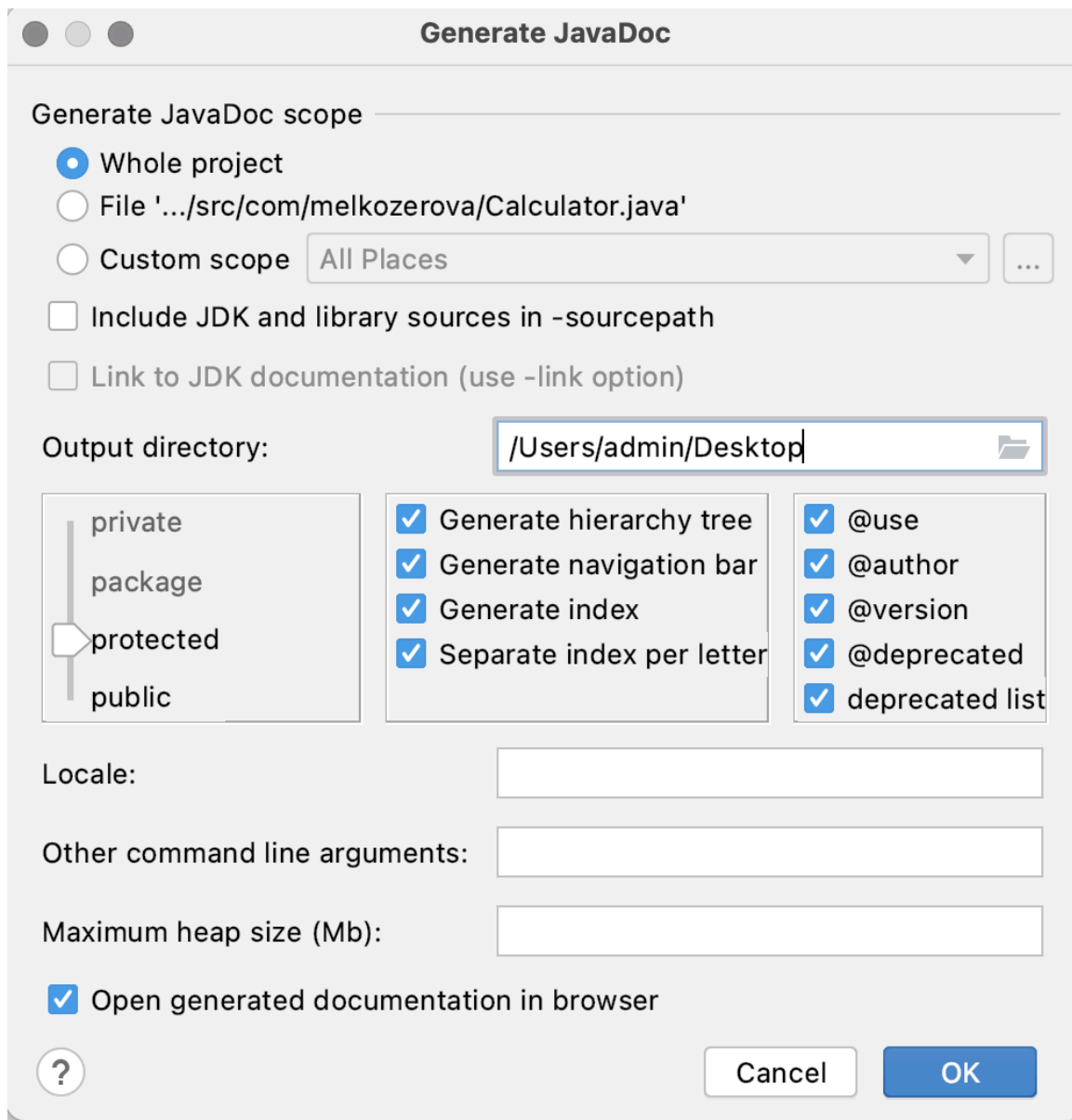


Рис. 2 Генерація Javadoc

```

Building tree for all the packages and classes...
/Users/admin/Documents/lab5_pais/src/com/melkozerova/Calculator.java:5: error: unknown tag: autor
* @autor Мелкозерова Ольга ДА-72
^
/Users/admin/Documents/lab5_pais/src/com/melkozerova/CalculatorUtest.java:10: error: unknown tag: autor
* @autor Мелкозерова Ольга ДА-72
^

```

Рис.3 Помилка через одруківку в теґу

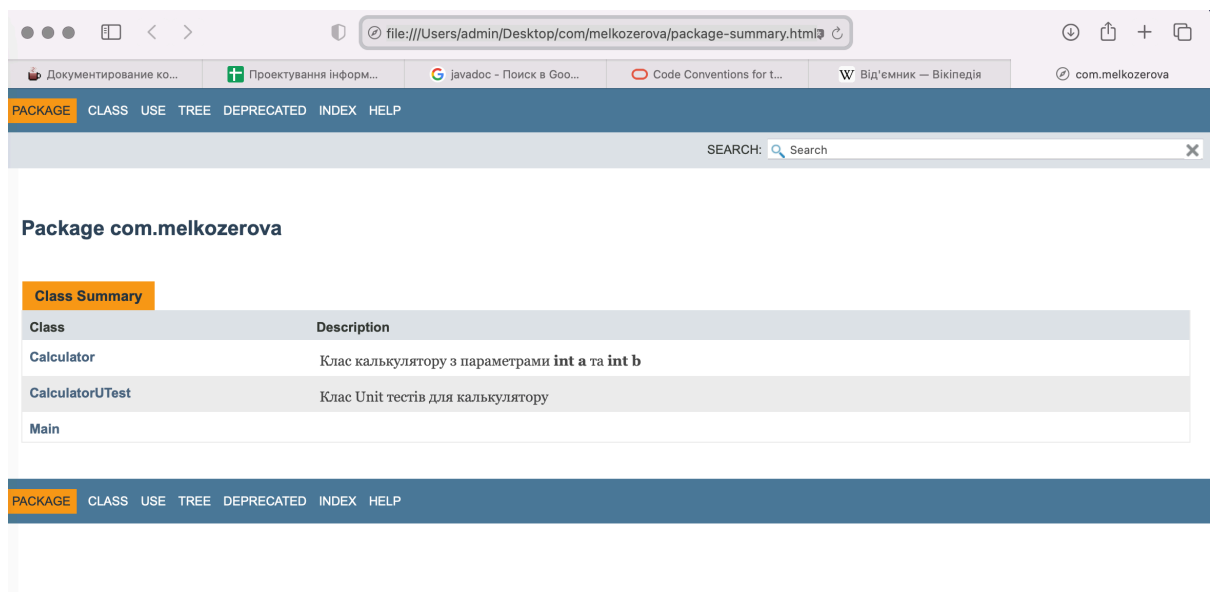


Рис. 4 Документація загальна

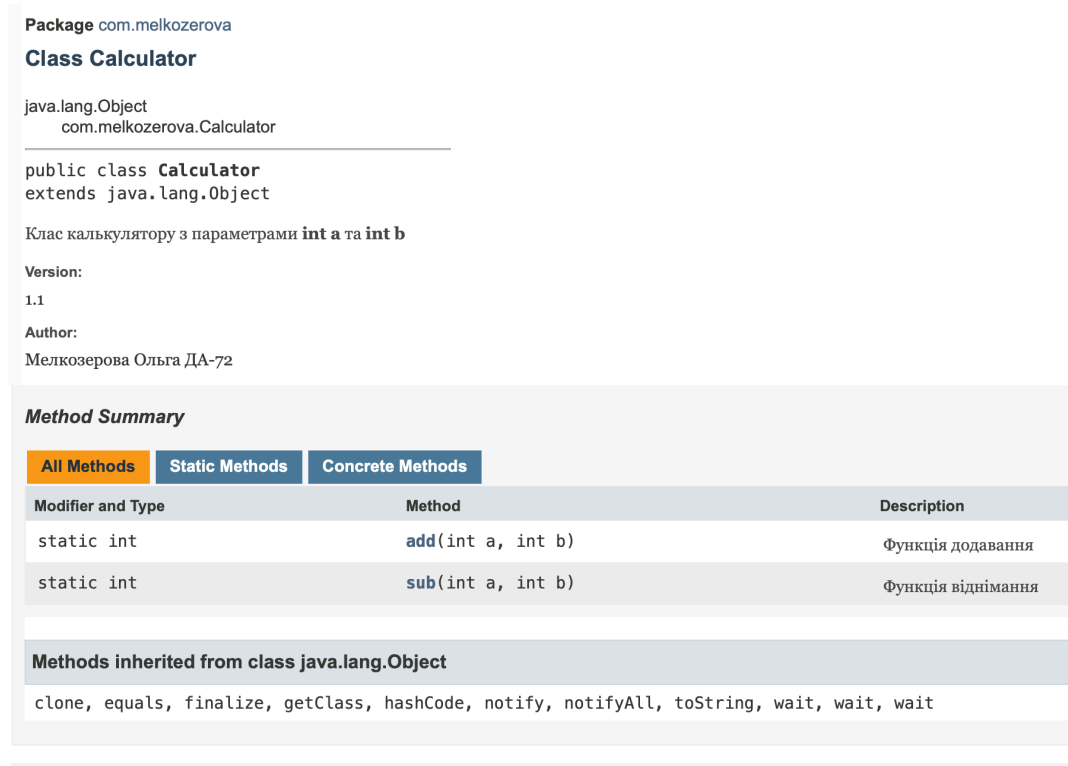


Рис. 5 Метод додавання

Висновок: у ході роботи був досліджений принцип автогенерації документації для мови Java за допомогою засобі JavaDoc, на виході була створена документація в форматі HTML. Складнощів під час виконання лабораторної роботи не виникало, адже це вбудований і інтуїтивно досить зрозумілий засіб для створення документації.