

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

Виконав:

студент групи ДА-72

Пономарьов Олег

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

Для виконання даної лабораторної роботи використаємо Javadoc. Javadoc це генератор документації в HTML, він також є стандартом документування.

Він вбудований в IntelliJ IDEA, тому ніяких додаткових підключень не потребується.

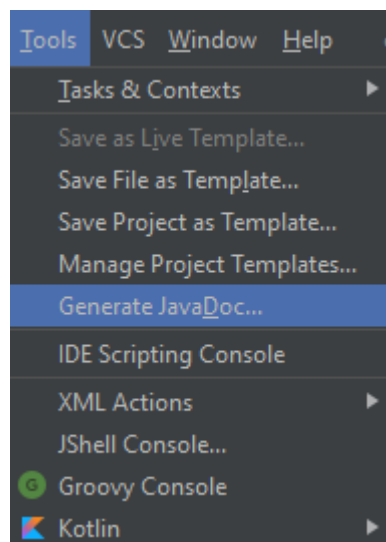



Рис.1 Стандартний засіб для генерації JavaDoc

Документування класу, методу чи змінної починається з комбінації символів `/**`, після якого слідує тіло коментарів, а закінчується все `*/`. В тіло коментарів можна вставляти

дескриптори, що починаються з `@` та нової строки.

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

```
/**
 *  @version 1.0
 * @author tor Ponomarev Oleh
 *
 * Даний клас реалізує метод с двома
 * целочисленными аргументи сложения двух чисел.
 *
 * Функция возвращает сумму двух аргументов.
 */
```

```
Generating G:\package-summary.html...
Generating G:\package-tree.html...
Generating G:\constant-values.html...
Generating G:\overview-tree.html...
Generating G:\deprecated-list.html...
Building index for all the packages and classes...
Generating G:\index-files\index-1.html...
Generating G:\index-files\index-2.html...
Generating G:\index-files\index-3.html...
Building index for all classes...
Generating G:\allclasses-index.html...
Generating G:\allpackages-index.html...
Generating G:\index.html...
Generating G:\help-doc.html...
```

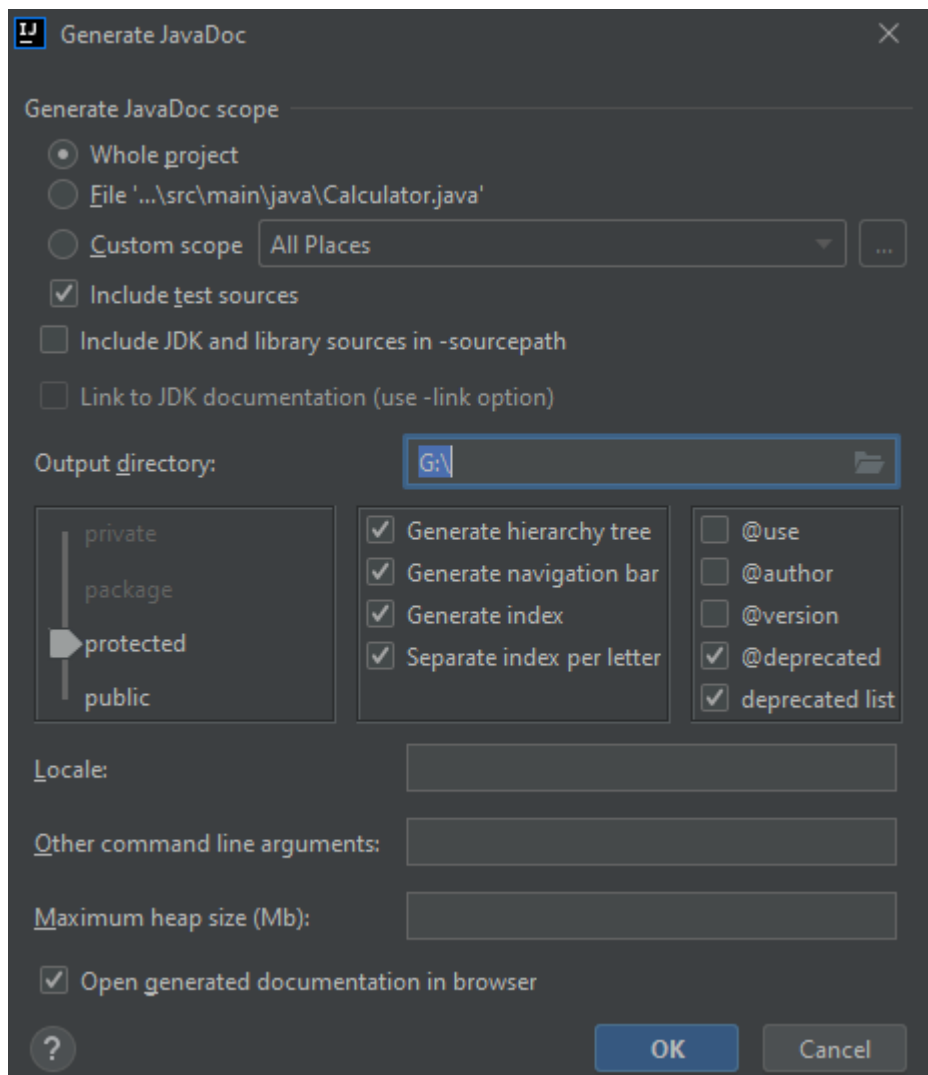


Рис. 2 Генерація Javadoc

PACKAGE

CLASS

TREE

DEPRECATED

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD

SEARCH

Class Calculator

java.lang.Object

Calculator

public class Calculator

extends java.lang.Object

Constructor Summary

Constructors

Constructor

Calculator()

Description

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type

Method

Description

int

add(int a, int b)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

Calculator

public Calculator()

Method Details

add

public int add(int a,

Рис. 4 Документація загальна

| |
|--|
| <div> <div>Constructor Details</div> <div> <div>Calculator</div> <div> <div>public Calculator()</div> </div> </div> </div> |
| <div> <div>Method Details</div> <div> <div>add</div> <div> <div>public int add(int a,</div> <div>int b)</div> </div> </div> </div> |

Рис. 5 Метод додавання

Наведемо приклади Coding Convention

Правильно:

```
import org.junit.Assert;
import org.junit.Test;

public class TestCalculator {
    @Test
    public void testAdd() throws Exception{
        Calculator calculator = new Calculator();
        int add = calculator.add( a: 3, b: 4);
        Assert.assertEquals( expected: 7, add);
    }
}
```

Неправильно:

```
public Class testCalculator {
    @Test
    public void testAdd() throws Exception{
        Calculator -calculator = new Calculator();
        int add = calculator.add(3, 4);
        Assert.assertEquals(7, add);
    }
}
```

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.