

**МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені Ігоря Сікорського»**  
**НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС**  
**«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»**  
**Кафедра Системного проектування**

Лабораторна робота №6  
з дисципліни: проектування інформаційних систем  
на тему «Створення документації»

Виконала  
Студентка 4 курсу  
групи ДА-71  
Опрісник Роксолана

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.

**Завдання**

- Для кожного з класів API в код програми додати теги з описом керівництва користувача для архітектурної програмної моделі.
- Обрати Coding Convention. Оформити код програми згідно Coding Convention.
- Встановити налаштування системи автоматичного створення керівництва користувача.
- Згенерувати HTML документацію керівництва користувача.
- Згенерувати PDF документацію керівництва користувача.

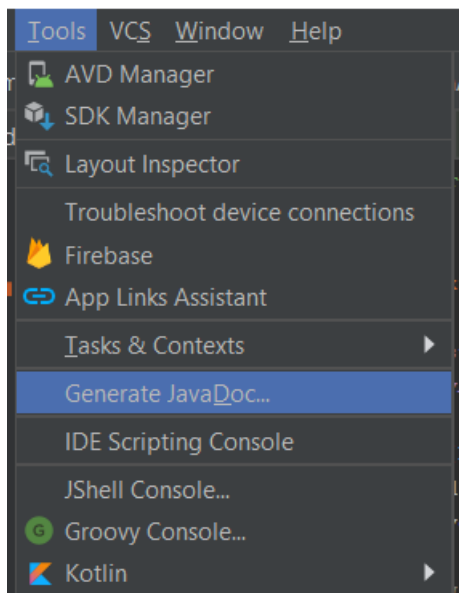
## Хід роботи

Для створення Coding Convention був обраний внутрішній інструмент Android studio для створення документації.

Використанні теги:

Дескриптор Параметр	Призначення	Використання	З версії
@author <i>Опис</i>	Ідентифікує автора.	Клас, інтерфейс	1.0
{@code <i>Фрагмент коду</i> }	Відображає інформацію «як є», без обробки HTML-стилю, проте з використанням шрифту коду.	Скрізь	1.5
@deprecated <i>Опис</i>	Описує застарілий клас або його член. При використанні бажано вказувати, чим було замінено функціональність.	Клас, інтерфейс, метод, змінна	1.0
{@docRoot}	Визначає шлях до корінного каталогу поточної директорії.	Скрізь	1.3
@exception <i>Назва виключення</i> <i>Пояснення</i> @throws <i>Назва виключення</i> <i>Пояснення</i>	Описує виключення, що може генерувати метод.	Метод	1.0 та 1.2 відповідно
{@inheritDoc}	Наслідує коментар безпосередньо від <a href="#">предка</a> .	Підмінений метод	1.4
{@link <i>Пакет. Клас#Член</i> <i>Опис</i> }	Вставляє вбудоване посилання на іншу тему.	Скрізь	1.2
{@linkplain}	Вставляє вбудоване посилання на іншу тему, проте відображає його у вигляді відкритої форми шрифту.	Скрізь	1.4
{@literal <i>Опис</i> }	Відображає інформацію «як є», без обробки HTML-стилю.	Скрізь	1.5
@param <i>Назва параметру</i> <i>Опис</i>	Документує параметр методу, класу чи інтерфейсу. При використанні не потрібно вказувати тип параметру.	Клас, інтерфейс, метод	1.0
@return <i>Опис</i>	Описує значення, що повертає метод.	Метод	1.0
@see <i>Тема</i> @see <i>Пакет. Клас#Член</i> <i>Опис</i>	Визначає посилання на іншу тему документації.	Клас, інтерфейс, метод, змінна	1.0
@serial <i>Опис</i>	Документує змінну, що серіалізується за замовчуванням.	Змінна	1.2
@serialData <i>Опис</i>	Документує дані, записані методами <code>writeObject()</code> та <code>writeExternal()</code> .	Метод	1.2
@serialField <i>Опис</i>	Документує компонент <code>ObjectStreamField</code> .	Змінна	1.2
@since <i>Версія</i>	Показує, коли було введено дану функціональність.	Скрізь	1.1
{@value} {@value <i>Пакет. Клас#Змінна</i> }	Повертає значення статичної змінної.	Статична змінна	1.4
@version <i>Версія</i>	Визначає версію класу і може використовуватися в ньому лише раз.	Клас, інтерфейс	1.0

## Напис



Generate Javadoc scope

☒ Whole project☐

Custom scope

Module 'app'

▼

...

☒

Include test sources

☐

Include JDK and library sources in -sourcepath

☐

Link to JDK documentation (use -link option)

Output directory:

D:\

☐

private

☐

package

☐

protected

☒

public

☒

Generate hierarchy tree

☒

Generate navigation bar

☒

Generate index

☒

Separate index per letter

☒

@use

☒

@author

☒

@version

☒

@deprecated

☒

deprecated list

Locale:

Other command line arguments:

Maximum heap size (Mb):

☒

Open generated documentation in browser

?

OK

Cancel

Документація до додатку:

```
class MainActivity extends AppCompatActivity {

    private ListView userList;
    private ArrayAdapter<Film> arrayAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        userList = (ListView) findViewById(R.id.list);

        userList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Film user = arrayAdapter.getItem(position);
                if (user != null) {
                    Intent intent = new Intent(getApplicationContext(), AddFilm.class);
                    intent.putExtra("name", user.getId());
                    intent.putExtra("click", value: 25);
                    startActivity(intent);
                }
            }
        });

    }

    @Override
    public void onResume() {
        super.onResume();
        DatabaseAdapter adapter = new DatabaseAdapter(this);
        adapter.open();
    }
}
```

Вихідний код

файл index.html

1. Вміст проекту

PACKAGE	CLASS	USE	TREE	DEPRECATED	INDEX	HELP
PREV PACKAGE	NEXT PACKAGE	FRAMES	NO FRAMES			
Package com.example.eatandtrain						
Class Summary						
Class	Description					
AddFilm	Implements the interface for adding, editing, and deleting a movie, and implements field validation.					
DatabaseAdapter	Database class, acts as a repository, and contains database queries					
DatabaseHelper	A class to help interact with the database					
ExampleInstrumentedTest	Instrumented test, which will execute on an Android device.					
ExampleUnitTest	Example local unit test, which will execute on the development machine (host).					
Film	A class representing the main fields and methods for accessing the fields					
MainActivity	Implements the record display interface as well as the ability to add new ones					
MainActivityTest						
PACKAGE	CLASS	USE	TREE	DEPRECATED	INDEX	HELP
PREV PACKAGE	NEXT PACKAGE	FRAMES	NO FRAMES			

## 2. Приклад

PREV CLASS NEXT CLASS FRAMES NO FRAMES  
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.example.filmoteka

**Class DatabaseAdapter**

java.lang.Object  
com.example.filmoteka.DatabaseAdapter

---

```
public class DatabaseAdapter
extends java.lang.Object
```

Database class, acts as a repository, and contains database queries

---

**Constructor Summary**

**Constructors**

Constructor and Description
DatabaseAdapter(android.content.Context context)

---

**Method Summary**

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
void	close()	
long	delete(long userId) Delete record from data base	
long	getCount()	
Film	getFilm(long id) Get a film which user want to see	

Напис

### Висновки:

Інструмент легкий в користуванні та не потребує додаткових ґрунтовних знань. Містить gui інтерфейс та може бути запущена прямо під час розробки. Базуючись на налаштуваннях, цей інструмент створює автоматично з source code відповідні. Основні теги Java-аннотацій використовуються для створення відповідної документації для проекту на Java через програму