

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 4
з дисципліни
«Розробка мобільних застосунків під Android»

Виконала:
Студентка 3 курсу ФІОТ
групи ІМ-22
Черникова Дар'я Дмитрівна

Перевірив:
Орленко Сергій Петрович

Київ 2025

Мета роботи: дослідити яким чином платформа Андроїд надає можливість оброблювати аудіо-файли та відео-файли та отримати практичні навички щодо використання інструментів відтворення медіа-даних.

ЗАВДАННЯ

БАЗОВЕ (12/20 балів). Написати програму під платформу Андроїд, яка має інтерфейс для запуску аудіо-файлів та відео-файлів. Мінімально інтерфейс має надавати можливість Програвати/Зупиняти/Призупиняти відтворення відео-файлу або аудіо-файлу, який зберігається у внутрішньому сховищі.

Опис програми

Програма представляє собою Android-додаток для роботи з медіафайлами, реалізований на мові Kotlin. Архітектурно додаток побудований на основі одного класу Activity (MainActivity), який інкапсулює логіку роботи з двома медіа-компонентами: MediaPlayer для аудіофайлів та VideoView для відеофайлів.

Технічно програма використовує наступні компоненти Android SDK:

- **MediaPlayer** – для низькорівневого управління аудіопотоком
- **VideoView** – компонент для відтворення відеоконтенту з інтегрованим UI
- **ActivityResultLauncher** з контрактом **ActivityResultContracts.StartActivityForResult()** – для асинхронного запуску системних селекторів файлів
- **MediaStore.Audio.Media.EXTERNAL_CONTENT_URI** та **MediaStore.Video.Media.EXTERNAL_CONTENT_URI** – URI контент-провайдерів для доступу до аудіо та відео
- **ContentResolver.query()** – для отримання метаданих медіафайлів (імен файлів)

Реалізація дозволів використовує адаптивний підхід до різних версій API Android:

- Для Android 13+ (API 33) – запитуються гранульовані дозволи **READ_MEDIA_AUDIO** та **READ_MEDIA_VIDEO**
- Для Android 12 та нижче – використовується загальний дозвіл **READ_EXTERNAL_STORAGE**

Управління життєвим циклом медіа-компонентів реалізоване з використанням:

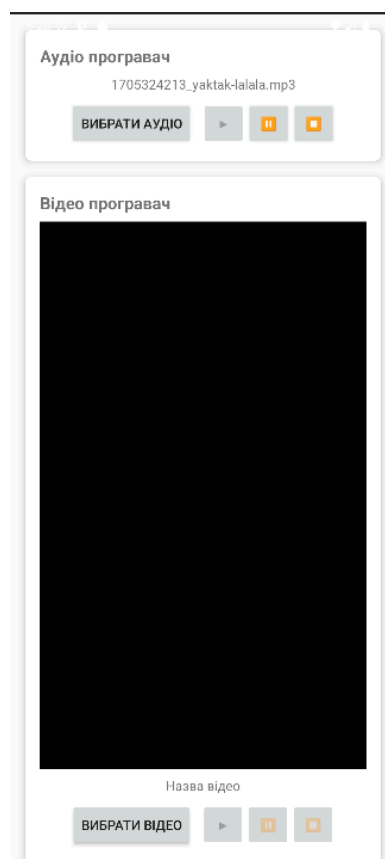
- Слухачів подій: `setOnCompleteListener` та `setOnErrorListener` для відстеження завершення та помилок відтворення
- Коректного звільнення ресурсів через виклик `release()` в методі `onDestroy()` для запобігання витоку пам'яті

Архітектура UI складається з двох блоків контролів, кожен з яких містить 4 кнопки (селектор, програвання, пауза, стоп) та текстове поле для відображення назви файлу. Стан UI динамічно змінюється в залежності від поточного стану відтворення через методи `updateAudioControlButtons()` та `updateVideoControlButtons()`, які керують доступністю (властивістю `isEnabled`) кнопок.

Функціональність програми

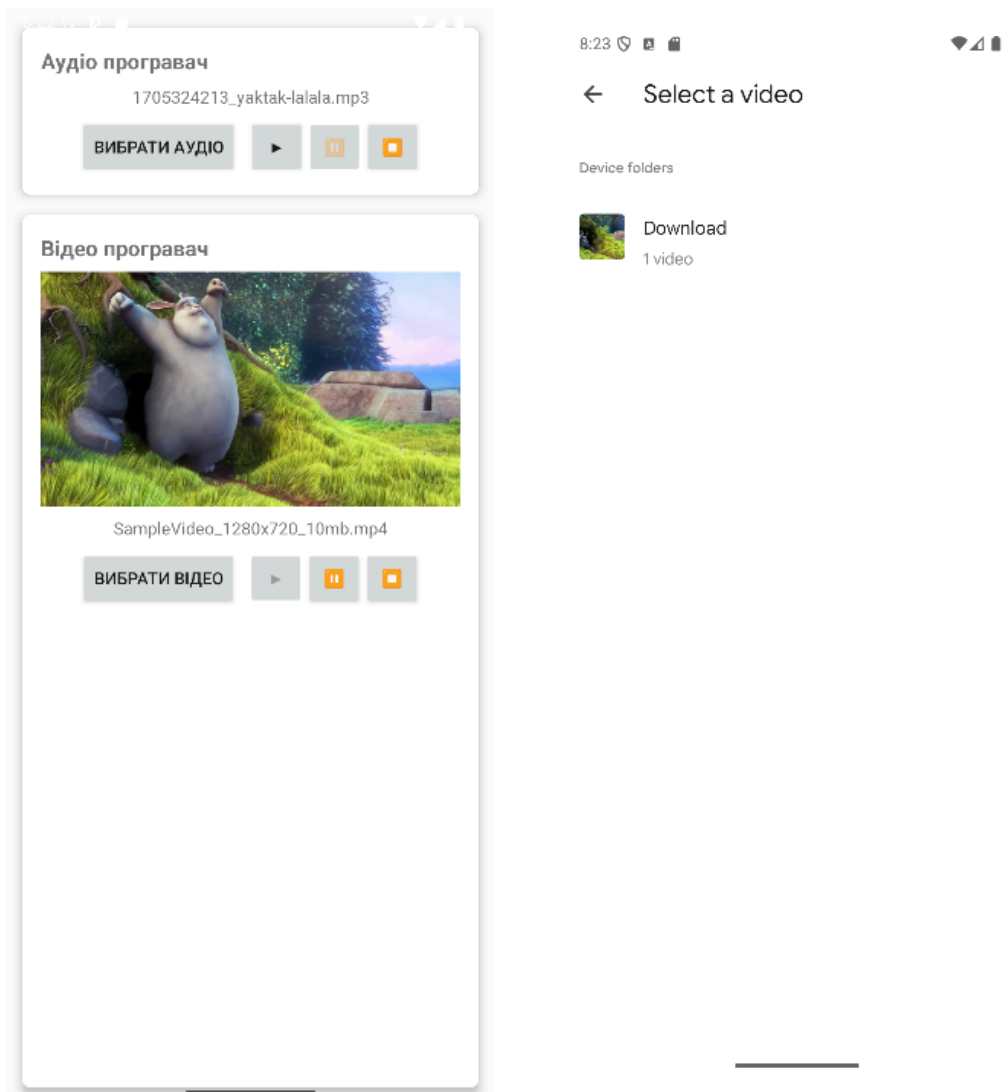
Управління аудіо:

- Вибір аудіофайлу з пам'яті пристрою
- Відтворення, пауза та зупинка аудіо
- Відображення назви вибраного аудіофайлу



Управління відео:

- Вибір відеофайлу з пам'яті пристрою
- Відтворення, пауза та зупинка відео
- Відображення назви вибраного відеофайлу
- Перегляд відео на екрані через компонент VideoView



Додаткові можливості:

- Автоматичне запитування необхідних дозволів для доступу до файлів
- Динамічна зміна стану кнопок в залежності від дій користувача
- Повідомлення про помилки при відтворенні файлів
- Автоматичне очищення ресурсів при виході з додатку