

ЛАБОРАТОРНА РОБОТА № 3

ДОСЛІДЖЕННЯ СПОСОБІВ ЗБЕРЕЖЕННЯ ДАНИХ

Мета роботи: дослідити способи збереження даних (база даних, файлова система, тощо) та отримати практичні навички щодо використання сховищ даних.

ЗАВДАННЯ

Написати програму під платформу Андроїд, яка доповнює програму, що розроблена за лабораторною роботою 2, роботою зі сховищами. Тобто при натисканні на кнопку «ОК» додатково:

- здійснюється запис результату взаємодії з інтерфейсом до сховища (файл або базу даних);
- користувач інформується відповідним повідомленням щодо успішності запису.

Також інтерфейс необхідно доповнити кнопкою «Відкрити», натискання на яку призводить до переходу на іншу Діяльність, у якій відображається вміст даних, що зберігаються у сховищі. Якщо дані відсутні (сховище пусте) відобразити відповідне повідомлення.

ОПИС ПРОГРАМИ

Функціонал:

Програма розширює функціонал лабораторної роботи №2, додаючи можливість збереження даних у файлову систему. Основні нововведення:

1. **Збереження даних:** При натисканні кнопки "ОК" у InputFragment, введені питання та відповіді зберігаються у текстовий файл questions_answers.txt з позначкою часу.

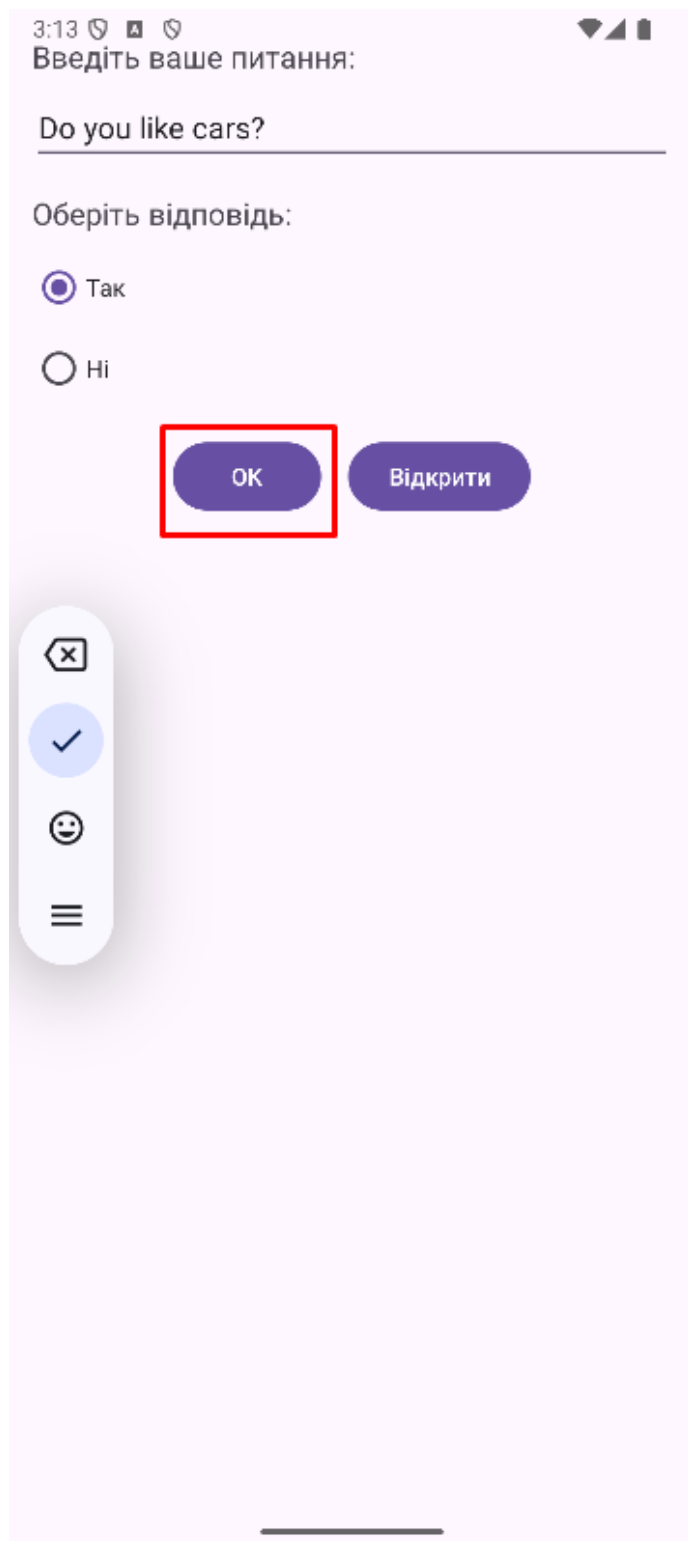


Рисунок 1 – Головне меню

2. **Кнопка "Відкрити"**: У InputFragment додана кнопка "Відкрити", яка відкриває нову активність FileViewActivity для перегляду збережених даних.

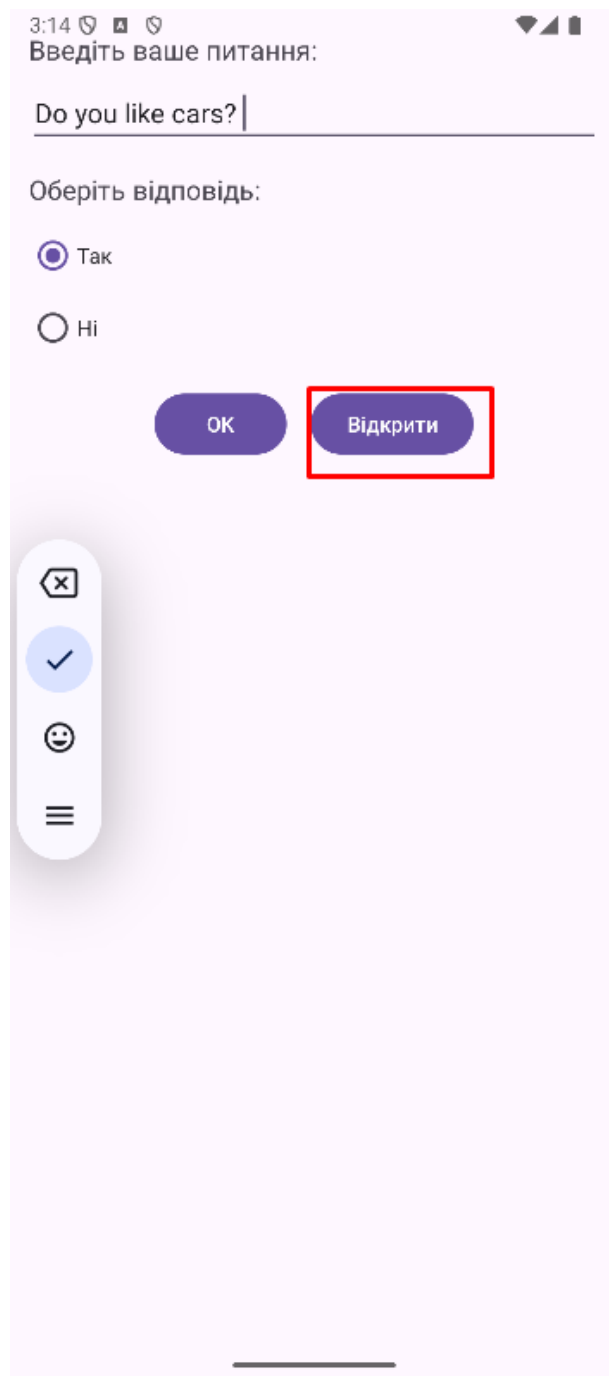


Рисунок 2 – Кнопка для відкриття збережених даних

3. **Перегляд збережених даних:** `FileViewActivity` відображає весь вміст файлу зі збереженими питаннями та відповідями.



Рисунок 3 – Збережені дані

4. **Очищення даних:** У FileViewActivity додана функція очищення файлу з підтвердженням через Toast-повідомлення.



Рисунок 4 – Очищення даних

5. **Обробка помилок:** Реалізована обробка помилок при читанні та записі файлів з відповідними повідомленнями користувачу.



Рисунок 5 – Повідомлення про помилку при очищенні даних

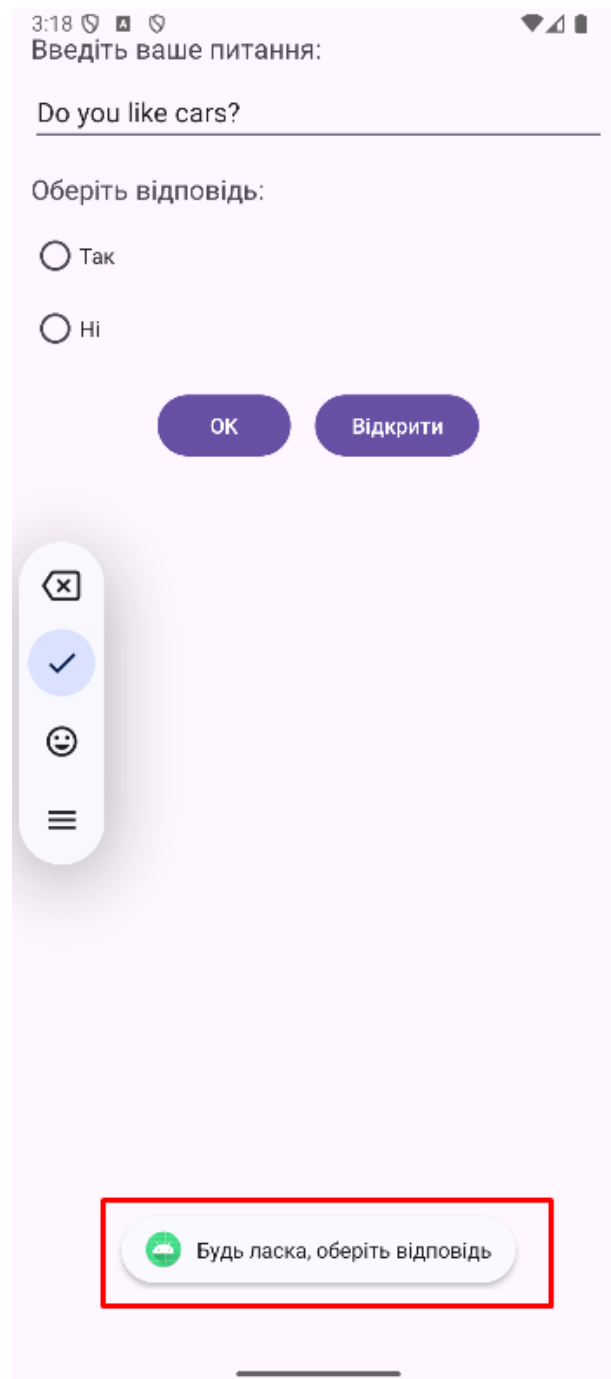


Рисунок 6 – Повідомлення про помилку при введенні даних

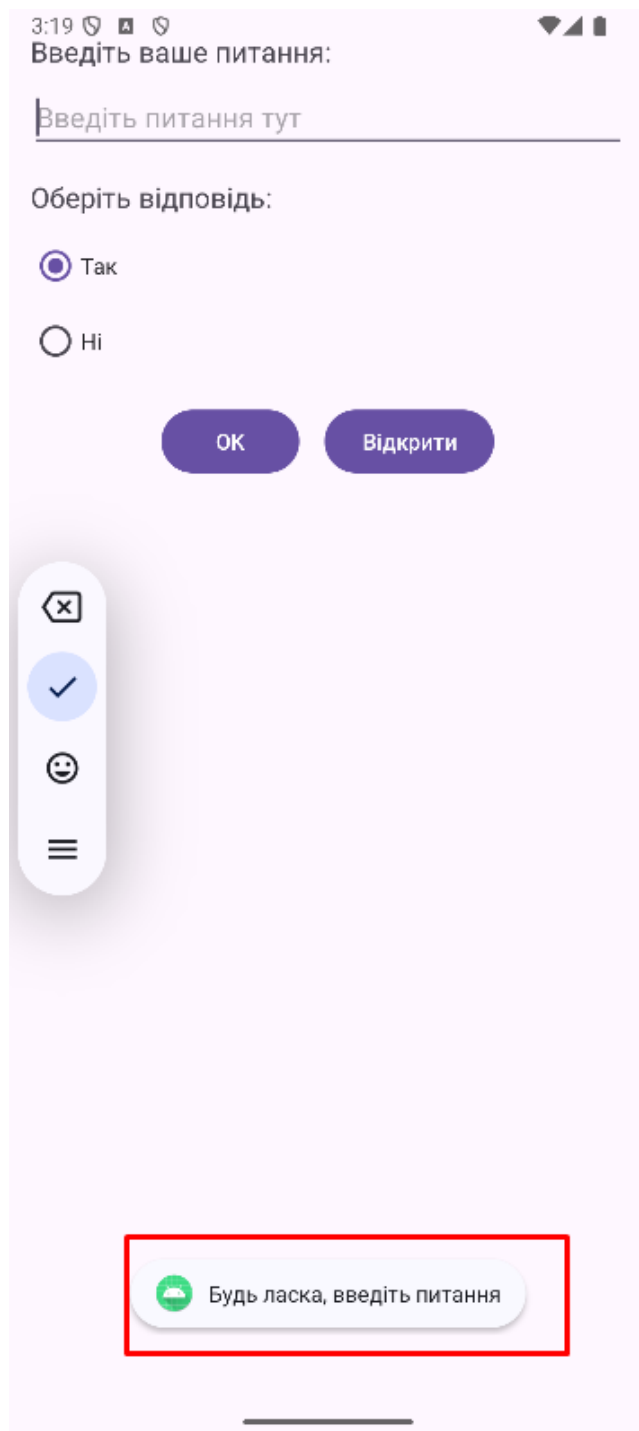


Рисунок 7 – Повідомлення про помилку при введенні даних

Структура програми

1. Головна активність (MainActivity.kt)

- Успадковує функціонал з лабораторної роботи №2
- Додано метод `saveToFile()` для збереження даних у файл

- Реалізовано додавання даних до існуючого файлу (append mode)
- Додано Toast-повідомлення про успішне збереження

2. Фрагмент введення даних (InputFragment.kt)

- Розширено функціонал попередньої версії
- Додана кнопка "Відкрити" (openButton)
- Реалізовано перехід до FileViewActivity через Intent
- Збережено валідацію введених даних

3. Фрагмент результату (ResultFragment.kt)

- Залишився без змін з лабораторної роботи №2
- Відображає введені питання та вибрану відповідь
- Кнопка "Скасувати" очищує форму та повертає до InputFragment

4. Активність перегляду файлів (FileViewActivity.kt)

- Нова активність для відображення збережених даних
- Метод loadFileContent() читає вміст файлу та відображає його
- Метод clearFileContent() видаляє файл з даними
- Кнопка "Назад" для повернення до головної активності
- Обробка випадків: файл не існує, файл порожній, помилки читання

Особливості реалізації:

Збереження даних:

- Використовується внутрішнє сховище додатку (filesDir)
- Кожен запис містить позначку часу у форматі "dd/MM/yyyy HH:mm:ss"
- Дані додаються до файлу без перезапису попередніх записів

Формат збереження:

[22/06/2025 14:30:25] Question: Do you like cars?| Answer: Так

[22/06/2025 14:31:10] Question: Do you like electric cars?| Answer: Ні

Обробка помилок:

- Try-catch блоки для всіх операцій з файлами

- Інформативні повідомлення про помилки
- Перевірка існування файлу перед операціями

КОНТРОЛЬНІ ПИТАННЯ

1. **Опишіть як організована робота з налаштуваннями (даними у вигляді пари «ключ-значення»).** Робота з налаштуваннями в Android організована через SharedPreferences - система збереження простих даних у форматі ключ-значення. SharedPreferences зберігає дані у XML файлах всередині додатку. Для отримання об'єкта використовується `getSharedPreferences()` або `getPreferences()`. Запис здійснюється через `SharedPreferences.Editor` з методами `putString()`, `putInt()`, `putBoolean()` та `commit()` або `apply()`. Читання відбувається через методи `getString()`, `getInt()`, `getBoolean()` з можливістю вказати значення за замовчуванням.
2. **Опишіть типи сховищ файлів та причини їх використання.** В Android існує два основні типи сховищ: внутрішнє та зовнішнє. Внутрішнє сховище (Internal Storage) - приватне для додатку, автоматично видаляється при деінсталяції, не потребує дозволів, використовується для конфіденційних даних. Зовнішнє сховище (External Storage) - може бути доступне іншим додаткам, потребує дозволів, файли можуть залишатися після видалення додатку, використовується для великих файлів та обмін даними між додатками.
3. **Опишіть процес роботи з файлами та файловою системою.** Робота з файлами включає створення, читання, запис та видалення файлів. Для внутрішнього сховища використовуються методи `openFileOutput()` і `openFileInput()`, або `File(filesDir, fileName)`. Для зовнішнього - `getExternalFilesDir()`. Операції виконуються через `FileInputStream/FileOutputStream` або використанням методів `File.readText()/File.writeText()`. Обов'язкова обробка `IOException` через try-catch блоки. Перевірка існування файлу через `File.exists()`.
4. **Опишіть процес роботи з базами даних SQLite за допомогою класу `SQLiteOpenHelper`, наведіть переваги та недоліки.** `SQLiteOpenHelper` - абстрактний клас для роботи з SQLite БД. Потрібно реалізувати `onCreate()` для створення таблиць та `onUpgrade()` для оновлення схеми. Методи `getReadableDatabase()` та `getWritableDatabase()` надають доступ до БД. Переваги: повний контроль над SQL, високу продуктивність, можливість складних запитів. Недоліки: багато boilerplate коду, ручне написання SQL

запитів, складність у підтримці, відсутність type-safety, ручна обробка курсорів.

5. **Як відбувається робота з БД за допомогою бібліотеки Room, наведіть переваги та недоліки.** Room - бібліотека ORM від Google для роботи з SQLite. Складається з Entity (таблиці), DAO (Data Access Object) та Database. Entity анотуються @Entity, DAO містять методи з SQL анотаціями (@Query, @Insert, @Update, @Delete), Database - абстрактний клас з @Database. Переваги: type-safety, менше boilerplate коду, автоматична генерація SQL, LiveData підтримка, міграції. Недоліки: додаткова залежність, обмеження у складних запитах, час компіляції збільшується.
6. **Наведіть характеристики екранів мобільних пристроїв.** Основні характеристики екранів мобільних пристроїв: розмір (діагональ у дюймах), роздільна здатність (кількість пікселів), щільність пікселів (PPI - pixels per inch), співвідношення сторін (16:9, 18:9, 19.5:9), тип матриці (LCD, OLED, AMOLED), яскравість (нітах), контрастність, кольоровий охоп, частота оновлення (60Hz, 90Hz, 120Hz), кути огляду, енергоспоживання.
7. **Наведіть класифікацію та відмінності технологій (типів) екранів мобільних пристроїв.** LCD (Liquid Crystal Display) - використовує рідкі кристали з підсвічуванням, дешевші, стабільні кольори, менший контраст. OLED (Organic Light-Emitting Diode) - кожен піксель світиться самостійно, глибокий чорний колір, високий контраст, енергоефективні для темних зображень. AMOLED (Active Matrix OLED) - покращена версія OLED з активною матрицею, кращі кути огляду. Super AMOLED - інтеграція сенсорного шару, тонші екрани. Retina - маркетингова назва Apple для високої щільності пікселів.
8. **Наведіть поняття та характеристики сенсорних екранів.** Сенсорний екран - пристрій введення інформації, який реагує на дотик пальця або стилуса. Основні характеристики: чутливість до дотику, підтримка мультитотику (Multi-touch), точність позиціонування, швидкість відгуку, стійкість до зносу, підтримка жестів (swipe, pinch, rotate), можливість роботи в рукавичках, захист від випадкових дотиків, калібрування.
9. **Наведіть загальну класифікацію сенсорних екранів.** За принципом роботи: резистивні (тиск на плівку), ємнісні (зміна електричної ємності), інфрачервоні (перерва світлового променя), ультразвукові (хвилі на

поверхні). За кількістю точок дотику: одnodотикові та мультитотикові. За технологією: проекційно-ємнісні (найпоширеніші в смартфонах), поверхнево-ємнісні, резистивні 4/5-провідні. За розміром: малі (смартфони), середні (планшети), великі (інтерактивні дошки).

10. Наведіть рекомендації щодо розробки інтерфейсів для сенсорних екранів. Основні рекомендації: мінімальний розмір інтерактивних елементів 44-48dp, достатня відстань між елементами (8dp), використання жестів (swipe, pinch-to-zoom), зворотній зв'язок при дотику (зміна кольору, анімація), адаптація під різні розміри екранів, оптимізація для однієї руки, використання Material Design принципів, тестування на реальних пристроях, підтримка accessibility, уникнення складних жестів для базового функціоналу.