



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №3
із дисципліни «Розробка мобільних застосунків під Android»
ДОСЛІДЖЕННЯ СПОСОБІВ ЗБЕРЕЖЕННЯ ДАНИХ

Виконала
студентка групи ІА–24
Сіденко Дар'я

Київ 2025

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 порожнє) відобразити відповідне повідомлення. За бажанням можна додатково реалізувати оновлення та видалення даних зі сховища.

Варіант: 138

Вікно містить текстове поле для введення деякого питання, дві опції (радіо батони) для відповіді (так або ні) та кнопку «ОК». Вивести введене питання і обрану відповідь у інше текстове поле.

Хід роботи

Для реалізації збереження даних у додатку були використані класи `FileOutputStream` та `FileInputStream` — основні інструменти для запису та зчитування даних у внутрішньому сховищі Android-додатку.

Зміни були внесені в усі основні класи проєкту. У класі `InputFragment` при натисканні кнопки «ОК» зчитується введене питання та обрана відповідь. Отримані дані зберігаються у файл за допомогою `FileOutputStream`. Кожен новий запис додається до файлу з нового рядка, що дозволяє формувати список збережених результатів.

У класі `OutputFragment` реалізовано зчитування збережених даних з файлу за допомогою `FileInputStream` та виведення їх у текстовому полі. Якщо файл

відсутній або порожній, виводиться відповідне повідомлення. Крім того, була додана кнопка «Видалити», при натисканні на яку виконується перевірка наявності файлу, і у разі його існування — видалення.

У MainActivity реалізовано логіку переходу між фрагментами та передачі керування при відкритті або приховуванні результатів.

Також були внесені зміни до інтерфейсу: додано нові кнопки та оновлено розмітку фрагментів, щоб забезпечити зручну роботу з функціональністю збереження та перегляду даних.

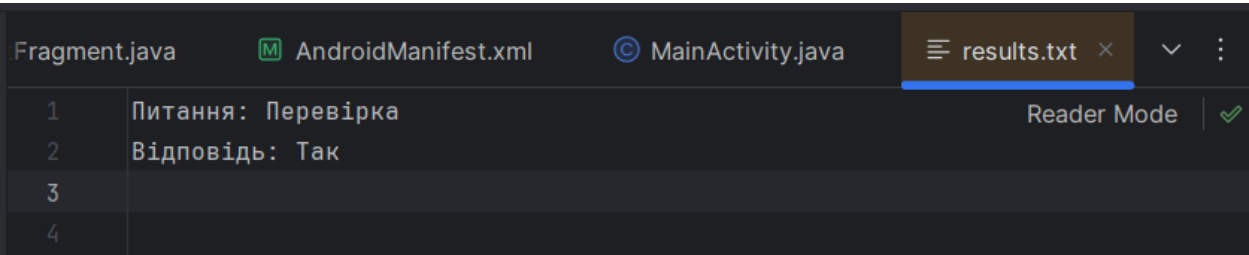


Рис.1 – Вміст файлу з даними

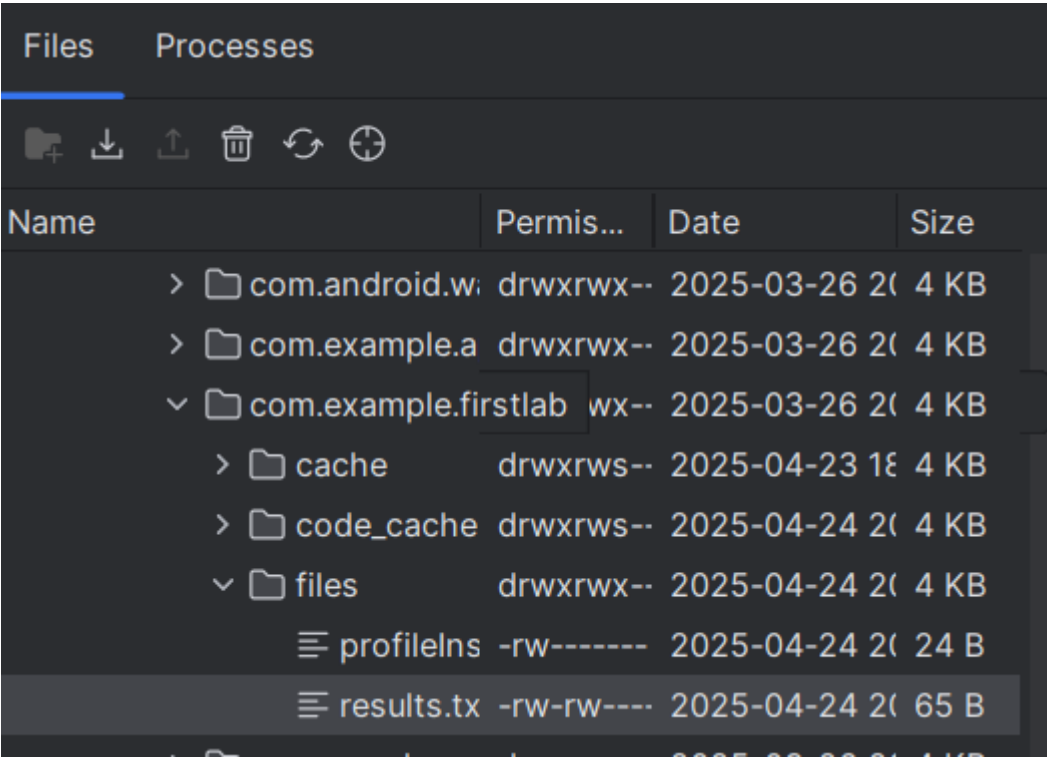


Рис.2 – Розташування файлу з даними у сховищі

Лістинг коду в *activity_main.xml*:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/fragment_container"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity" />
```

Відповідно самі фрагменти знаходяться в окремих файлах з розміткою.

Лістинг коду в fragment_input.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/editQuestion"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:minHeight="48dp"
        android:hint="Введіть питання"
        android:inputType="text"/>

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:orientation="horizontal">

        <RadioButton
            android:id="@+id/radioYes"
            android:layout_width="96dp"
            android:layout_height="wrap_content"
            android:text="Так" />

        <RadioButton
            android:id="@+id/radioNo"
            android:layout_width="83dp"
            android:layout_height="wrap_content"
            android:text="Ні" />
    </RadioGroup>

    <Button
        android:id="@+id/buttonOk"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="ОК" />

    <Button
        android:id="@+id/buttonOpen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Відкрити історію" />
</LinearLayout>
```

Лістинг коду в fragment_output.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/resultText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textAlignment="center"
        android:layout_marginBottom="16dp" />

    <Button
        android:id="@+id/buttonCancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cancel" />
</LinearLayout>
```

Лістинг коду в activity_file_view.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:padding="8dp">

        <TextView
            android:id="@+id/textFileContent"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="16sp" />
    </ScrollView>

    <Button
        android:id="@+id/buttonDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Видалити дані" />

    <Button
        android:id="@+id/buttonBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Повернутись" />
</LinearLayout>
```

Основні зміни *MainActivity.java*:

Оновлено метод `showOutputFragment()` таким чином, що тепер він показує ще й фрагмент із збереженими записами.

Лістинг коду *MainActivity.java*:

```
public class MainActivity extends AppCompatActivity {
    private InputFragment inputFragment;

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

        inputFragment = new InputFragment();

        getSupportFragmentManager().beginTransaction()
            .add(R.id.fragment_container, inputFragment)
            .commit();
    }

    public void showOutputFragment(String question, String answer) {
        String result = "Питання: " + question + "\nВідповідь: " + answer;

        boolean saved = saveToFile(result);
        Toast.makeText(this, saved ? "Результат збережено" : "Помилка збереження", Toast.LENGTH_SHORT).show();

        OutputFragment outputFragment = OutputFragment.newInstance(question, answer);

        getSupportFragmentManager().beginTransaction()
            .replace(R.id.fragment_container, outputFragment)
            .addToBackStack(null)
            .commit();
    }

    public void hideOutputFragment() {
        getSupportFragmentManager().popBackStack();
    }

    private boolean saveToFile(String data) {
        try {
            FileOutputStream fos = openFileOutput("results.txt",
MODE_APPEND);
            fos.write((data + "\n\n").getBytes());
            fos.close();
            return true;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

Основні зміни в *InputFragment.java*:

За допомогою `FileOutputStream` реалізовано збереження введеного тексту та

вибраної відповіді у файл. Додано повідомлення Toast про успішне збереження або помилку запису. Таким чином, фрагмент тепер не тільки передає дані іншому фрагменту, а й зберігає їх у внутрішньому сховищі.

Лістинг коду *InputFragment.java*:

```
public class InputFragment extends Fragment {
    private EditText editQuestion;
    private RadioGroup radioGroup;

    public InputFragment() {}

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_input, container,
false);

        editQuestion = view.findViewById(R.id.editQuestion);
        radioGroup = view.findViewById(R.id.radioGroup);
        Button buttonOk = view.findViewById(R.id.buttonOk);
        Button buttonOpen = view.findViewById(R.id.buttonOpen);

        buttonOk.setOnClickListener(v -> {
            String question = editQuestion.getText().toString().trim();
            int selectedId = radioGroup.getCheckedRadioButtonId();

            if (question.isEmpty() || selectedId == -1) {
                Toast.makeText(getActivity(), "Будь ласка, заповніть всі
поля", Toast.LENGTH_SHORT).show();
                return;
            }

            RadioButton selectedRadio = view.findViewById(selectedId);
            String answer = selectedRadio.getText().toString();

            if (getActivity() instanceof MainActivity) {
                ((MainActivity) getActivity()).showOutputFragment(question,
answer);
            }
        });

        buttonOpen.setOnClickListener(v -> {
            Intent intent = new Intent(getActivity(),
FileViewActivity.class);
            startActivity(intent);
        });

        return view;
    }

    @Override
    public void onResume() {
        super.onResume();
        clearForm();
    }

    public void clearForm() {
        if (editQuestion != null) {
            editQuestion.setText("");
        }
    }
}
```

```

    }
    if (radioGroup != null) {
        radioGroup.clearCheck();
    }
}
}

```

Основні зміни *OutputFragment.java*:

За допомогою `FileInputStream` читається весь вміст файлу і відображається у `TextView`. Якщо файл відсутній або порожній, виводиться повідомлення про відсутність збережених відповідей. Додано кнопку “Видалити” та реалізовано її функціонал.

Лістинг коду *OutputFragment.java*:

```

public class OutputFragment extends Fragment {
    private static final String ARG_QUESTION = "question";
    private static final String ARG_ANSWER = "answer";

    public OutputFragment() {}

    public static OutputFragment newInstance(String question, String answer)
    {
        OutputFragment fragment = new OutputFragment();
        Bundle args = new Bundle();
        args.putString(ARG_QUESTION, question);
        args.putString(ARG_ANSWER, answer);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_output, container,
false);

        TextView resultText = view.findViewById(R.id.resultText);
        Button buttonCancel = view.findViewById(R.id.buttonCancel);

        if (getArguments() != null) {
            String question = getArguments().getString(ARG_QUESTION);
            String answer = getArguments().getString(ARG_ANSWER);
            resultText.setText("Питання: " + question + "\nВідповідь: " +
answer);
        }

        buttonCancel.setOnClickListener(v -> {
            if (getActivity() instanceof MainActivity) {
                ((MainActivity) getActivity()).hideOutputFragment();
            }
        });

        return view;
    }
}

```


Лістинг коду *FileViewActivity.java*:

```
public class FileViewActivity extends AppCompatActivity {

    private TextView fileContent;

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

        fileContent = findViewById(R.id.textFileContent);
        Button buttonDelete = findViewById(R.id.buttonDelete);
        Button buttonBack = findViewById(R.id.buttonBack);

        loadFileContent();

        buttonDelete.setOnClickListener(v -> {
            deleteFile("results.txt");
            Toast.makeText(this, "Файл очищено", Toast.LENGTH_SHORT).show();
            fileContent.setText("Дані відсутні.");
        });

        buttonBack.setOnClickListener(v -> finish());
    }

    private void loadFileContent() {
        try {
            FileInputStream fis = openFileInput("results.txt");
            BufferedReader reader = new BufferedReader(new
InputStreamReader(fis));
            StringBuilder sb = new StringBuilder();
            String line;

            while ((line = reader.readLine()) != null) {
                sb.append(line).append("\n");
            }
            reader.close();

            String content = sb.toString().trim();
            fileContent.setText(content.isEmpty() ? "Дані відсутні." :
content);

        } catch (IOException e) {
            fileContent.setText("Файл не знайдено або пустий.");
        }
    }
}
```

Останній етап – тестування.

Випадок 1: відсутні записи.

Випадок 2: внесено запис.

Випадок 3: Видалення записів.

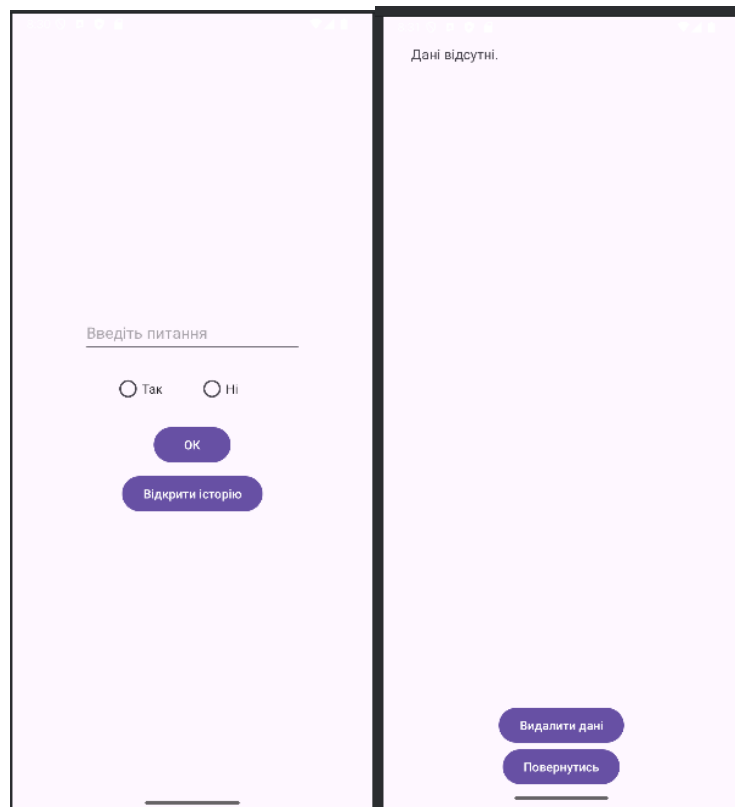


Рис.3 – Відсутні записи

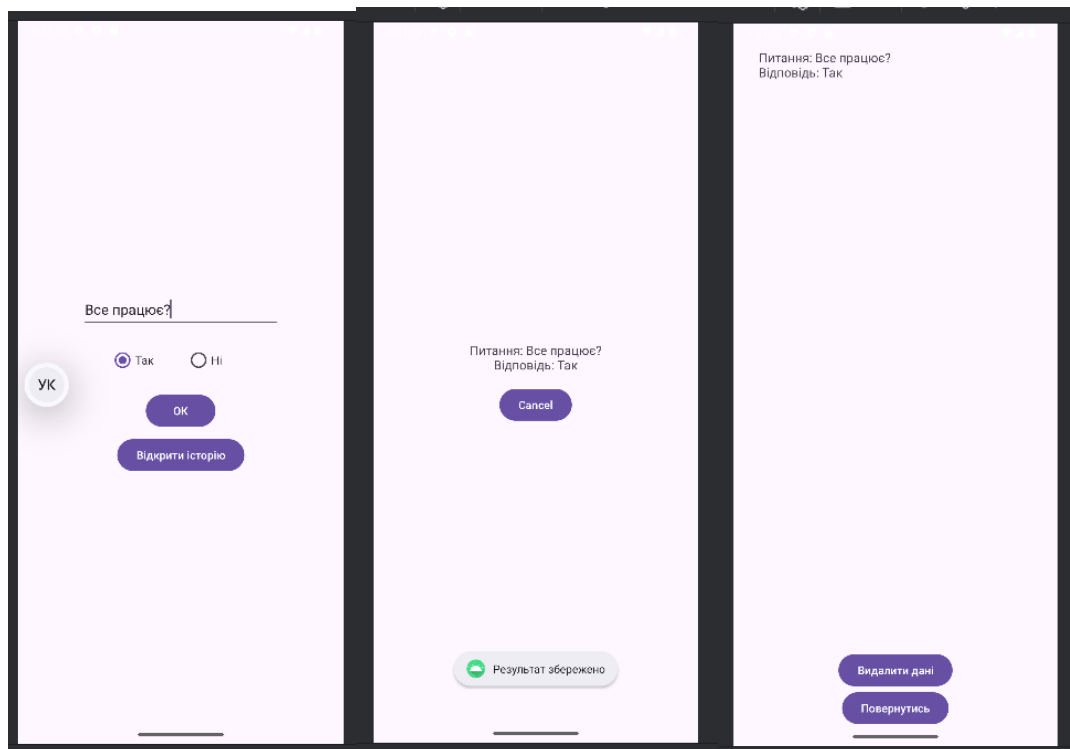


Рис. 4 - Додано запис

Відео з тестом програми за [посиланням](#).

Висновок: під час виконання даної лабораторної роботи я ознайомила з способами збереження даних та реалізувала один із них.