Практична робота 9. Робота зі сховищами даних в Android Studio

Теоретичні відомості

ОС Android побудовано на основі Linux. Цей факт знаходить своє відображення у роботі з файлами. Так, у шляхах до файлів як розмежувач в Linux використовує сліш "/", а не зворотний сліш "\" (як у Windows). А всі назви файлів і каталогів є реєстрозалежними, тобто "data" це не те саме, що і "Data".

Програма Android зберігає свої дані в каталозі /data/data/<назва_пакета>/ і, як правило, щодо цього каталогу буде йти робота.

Для роботи з файлами абстрактний клас android.content.Context визначає ряд методів:

- boolean deleteFile (String name): видаляє певний файл
- String[](): отримує посилання на підкаталог cache у каталозі програми
- File getDir(String dirName, int mode): отримує посилання на підкаталог у каталозі програми, якщо такого підкаталогу немає, то він створюється
- File getExternalCacheDir(): отримує посилання на папку /cache зовнішньої файлової системи пристрою
- File getExternalFilesDir(String type): отримує посилання на каталог /files зовнішньої файлової системи пристрою
- File getFileStreamPath(String filename): повертає абсолютний шлях до файлу у файловій системі
- FileInputStream openFileInput(String filename): відкриває файл
- FileOutputStream openFileOutput (String name, int mode): відкриває файл для запису

Усі файли, які створюються та редагуються в програмі, зазвичай зберігаються в підкаталозі /files в каталозі програми.

Для безпосереднього читання та запису файлів застосовуються також стандартні класи Java із пакету java.io.

Отже, застосуємо функціонал читання-запису файлів у додатку. Нехай у нас буде наступна примітивна розмітка layout:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/editor"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:textSize="18sp"
        android:gravity="start"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toTopOf="@id/save_text"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/save_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:onClick="saveText"
        android:text="Сохранить"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toTopOf="@id/text"
        app:layout_constraintTop_toBottomOf="@id/editor" />

    <TextView
        android:id="@+id/text"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:gravity="start"
        android:textSize="18sp"

        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/open_text"
        app:layout_constraintTop_toBottomOf="@+id/save_text"
/>
    <Button
        android:id="@+id/open_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="openText"
        android:text="Открыть"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Поле EditText призначене для введення тексту, а TextView - для виведення раніше збереженого тексту. Для збереження та відновлення тексту додано дві кнопки.

Тепер у коді Activity пропишемо обробники кнопок із збереженням та читанням файлу:

```
package com.example.filesapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {
     private final static String FILE_NAME = "content.txt";
     @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    // сохранение файла
    public void saveText(View view){

        FileOutputStream fos = null;
        try {
            EditText textBox = (EditText)
findViewById(R.id.editor);
            String text = textBox.getText().toString();
             fos = openFileOutput(FILE_NAME, MODE_PRIVATE);
            fos.write(text.getBytes());
            Toast.makeText(this, "Файл сохранен",
Toast.LENGTH_SHORT).show();
        }
        catch(IOException ex) {
            Toast.makeText(this, ex.getMessage(),
Toast.LENGTH_SHORT).show();
        }
```

```
        finally{
            try{
                if(fos!=null)
                    fos.close();
            }
            catch(IOException ex){
                Toast.makeText(this, ex.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        }
    }
    // открытие файла
    public void openText(View view){

        FileInputStream fin = null;
        TextView textView = (TextView)
findViewById(R.id.text);
        try {
            fin = openFileInput(FILE_NAME);
            byte[] bytes = new byte[fin.available()];
            fin.read(bytes);
            String text = new String (bytes);
            textView.setText(text);
        }
        catch(IOException ex) {
             Toast.makeText(this, ex.getMessage(),
Toast.LENGTH_SHORT).show();
        }
        finally{

            try{
                if(fin!=null)
                    fin.close();
            }
            catch(IOException ex){

                Toast.makeText(this, ex.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

При натисканні на кнопку збереження буде створюватись потік виводу FileOutputStream fos = openFileOutput(FILE_NAME, MODE_PRIVATE)

У цьому випадку введений текст зберігатиметься у файл "content.txt". При цьому використовуватиметься режим MODE_PRIVATE

Система дозволяє створювати файли з двома різними режимами:

– MODE_PRIVATE: файли можуть бути доступні тільки власнику програми (режим за замовчуванням)
– MODE_APPEND: дані можуть бути додані в кінець файлу

Тому в даному випадку якщо файл "content.txt" вже існує, то він буде перезаписаний. Якщо нам треба було дописати файл, тоді треба було б використовувати режим MODE_APPEND:

FileOutputStream fos = openFileOutput (FILE_NAME, MODE_APPEND);

Для читання файлу застосовується потік введення FileInputStream:

FileInputStream fin = openFileInput(FILE_NAME);

У результаті після натискання кнопки збереження весь текст буде збережено у файлі /data/data/назва_пакета/files/content.txt

Порядок виконання роботи

1. Вивчити методичні вказівки до практичного заняття.
2. Запустити на виконання Android Studio.
3. Створити у середовищі Android Studio проект, розроблений практичному занятті.
4. Додати до проекту елементи, які забезпечують роботу з файлами відповідно до індивідуального завдання.
5. Запустити створену програму в емуляторі Android і спостерігати за появою цієї програми та результатів його роботи у вікні додатків емулятора.

Зміст звіту

1. Назва та мета роботи.
2. Лістинг створеної програми в Android Studio.
3. Результати роботи програми в емуляторі телефону.