

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 №2

з дисципліни
"Розробка мобільних застосувань
під Android"
Тема: Дослідження роботи з
компонентом Fragment
Варіант 10 (номер за списком - 27)

Виконав:
студент групи IA-32
Діденко Я.О.

Перевірив:
Орленко С.П.

Мета роботи: дослідити створення та взаємодію з компонентом Фрагмент (Fragment) компоненту Діяльність та набути практичні навички з використання фрагментів для інтерфейсу користувача.

Завдання

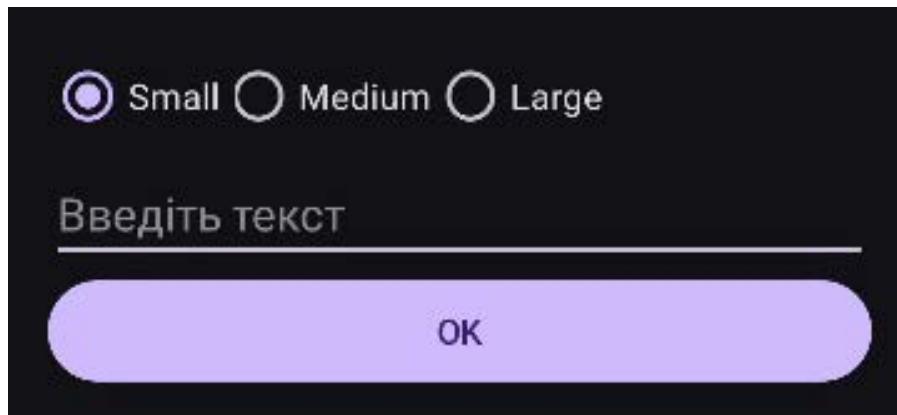
- | | |
|-----|--|
| 10. | Вікно містить групу опцій (радіо-батони) для вибору розміру шрифту, текстове поле для введення інформації та кнопки «OK» і «Cancel». Вивести введений рядок обраним розміром при натисканні на кнопку «OK» у деяке текстове поле та очистити при натисканні кнопки «Cancel». |
|-----|--|

Написати програму під платформу Андроїд, яка має інтерфейс, побудований з декількох фрагментів згідно варіанту. Перший фрагмент представляє з себе форму для введення даних та кнопку підтвердження («OK»), а інший фрагмент відображає результат взаємодії. Тобто другий фрагмент містить тестове поле з результатом та кнопкою «Cancel» (якщо згідно варіанту така існує, якщо ж за варіантом її немає – можете додати за власним бажанням), яка очищає або приховує (або видаляє) другий фрагмент та очищає форму введення з першого фрагменту. Зверніть увагу, що робота з фрагментами відбувається в рамках однієї Діяльності.

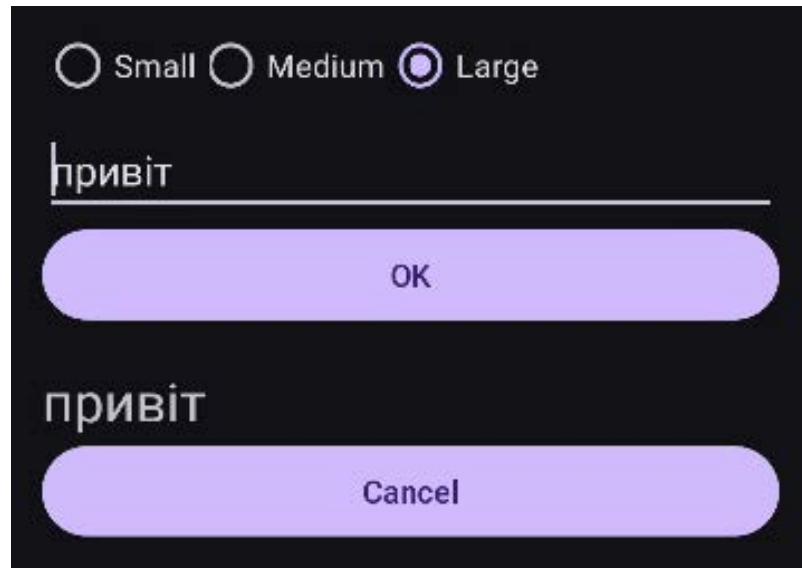
Примітка 1: завдання відповідає варіанту лабораторної роботи №1.

Хід роботи

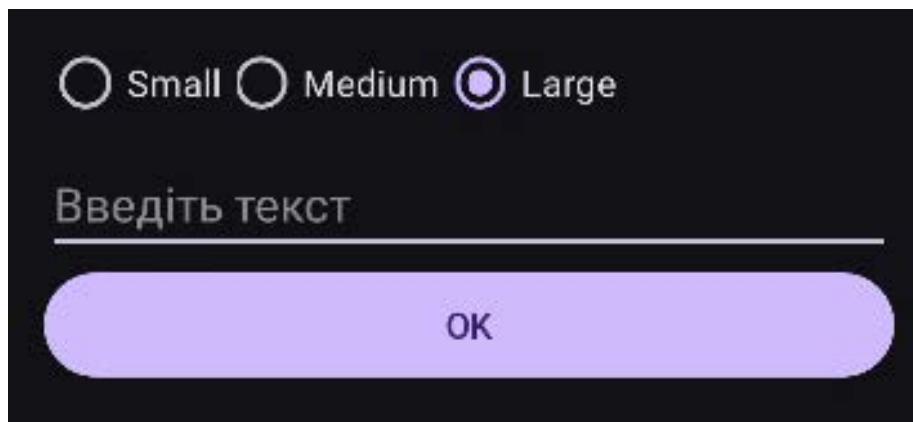
1. Бачимо інтерфейс програми. Інтерфейс складається з першого фрагмента (InputFragment), який містить групу радіо-кнопок для вибору розміру шрифту (Small, Medium, Large), текстове поле для введення інформації та кнопку «OK». Другий фрагмент на цьому етапі відсутній, оскільки він додається динамічно лише після підтвердження введених даних.



2. Вводимо текст та обираємо розмір шрифту. У текстове поле вводимо довільний рядок і обираємо один із запропонованих варіантів розміру. Після натискання кнопки «OK» введені дані передаються до ViewModel, у межах тієї ж Activity додається другий фрагмент (ResultFragment), у якому відображається введений текст із відповідним розміром шрифту.



3. Натискаємо кнопку «Cancel». Після натискання кнопки «Cancel» дані у ViewModel очищаються, текстове поле у першому фрагменті стає порожнім, а другий фрагмент видаляється з контейнера, внаслідок чого інтерфейс повертається до початкового стану.



4. Як саме все реалізовано? Взаємодія між фрагментами побудована через спільну ViewModel (UiViewModel), яка зберігає введений текст та вибраний розмір шрифту у змінних типу LiveData. Перший фрагмент - InputFragment отримує посилання на цю ViewModel через activityViewModels(), читає введений користувачем текст і вибраний розмір, після чого передає ці значення у ViewModel методом setData(). Далі в межах тієї ж Activity через FragmentManager динамічно додається другий фрагмент (ResultFragment). Другий фрагмент також підключається до тієї ж ViewModel і спостерігає за змінами даних через observe(), тому після оновлення значень автоматично відображає текст із відповідним розміром шрифту. Кнопка «Cancel» викликає метод clear() у ViewModel та видаляє ResultFragment, що призводить до очищення поля введення і повернення інтерфейсу до початкового стану без прямої передачі даних між фрагментами.

```
1 package com.example.lab2
2
3 import androidx.lifecycle.LiveData
4 import androidx.lifecycle.MutableLiveData
5 import androidx.lifecycle.ViewModel
6
7
8     2 Usages
9 class UiViewModel : ViewModel() {
10
11     3 Usages
12     private val _text = MutableLiveData( value = "" )
13     2 Usages
14     val text: LiveData<String> = _text
15
16     3 Usages
17     private val _fontSp = MutableLiveData( value = 18f )
18     1 Usage
19     val fontSp: LiveData<Float> = _fontSp
20
21     1 Usage
22     fun setData(text: String, fontSp: Float) {
23         _text.value = text
24         _fontSp.value = fontSp
25     }
26
27     1 Usage
28     fun clear() {
29         _text.value = ""
30         _fontSp.value = 18f
31     }
32 }
```

Висновки: У ході виконання лабораторної роботи було досліджено принцип створення та використання компонентів Fragment у межах однієї Activity. Реалізовано застосунок, інтерфейс якого складається з двох фрагментів: перший фрагмент призначений для введення даних і вибору параметрів, другий - для відображення результату. Було реалізовано динамічне додавання та видалення фрагмента за допомогою FragmentManager.

Для передачі та збереження стану даних використано ViewModel і LiveData, що дозволило організувати спільний стан між фрагментами без прямої залежності одного від одного. Забезпечено коректну взаємодію компонентів у межах однієї Activity та реалізовано механізм очищення даних і повернення інтерфейсу до початкового стану.

У результаті роботи набуті практичні навички створення фрагментів, керування їх життєвим циклом, організації взаємодії між ними та використання архітектурного підходу з ViewModel для збереження стану даних.