

# **ЛАБОРАТОРНА РОБОТА № 1**

## **ДОСЛІДЖЕННЯ РОБОТИ З ЕЛЕМЕНТАМИ КЕРУВАННЯ**

**Мета роботи:** дослідити створення простого застосунку під платформу Андроїд та набути практичні навички з використання елементів керування інтерфейсу, мов програмування Java чи Kotlin.

### **Завдання**

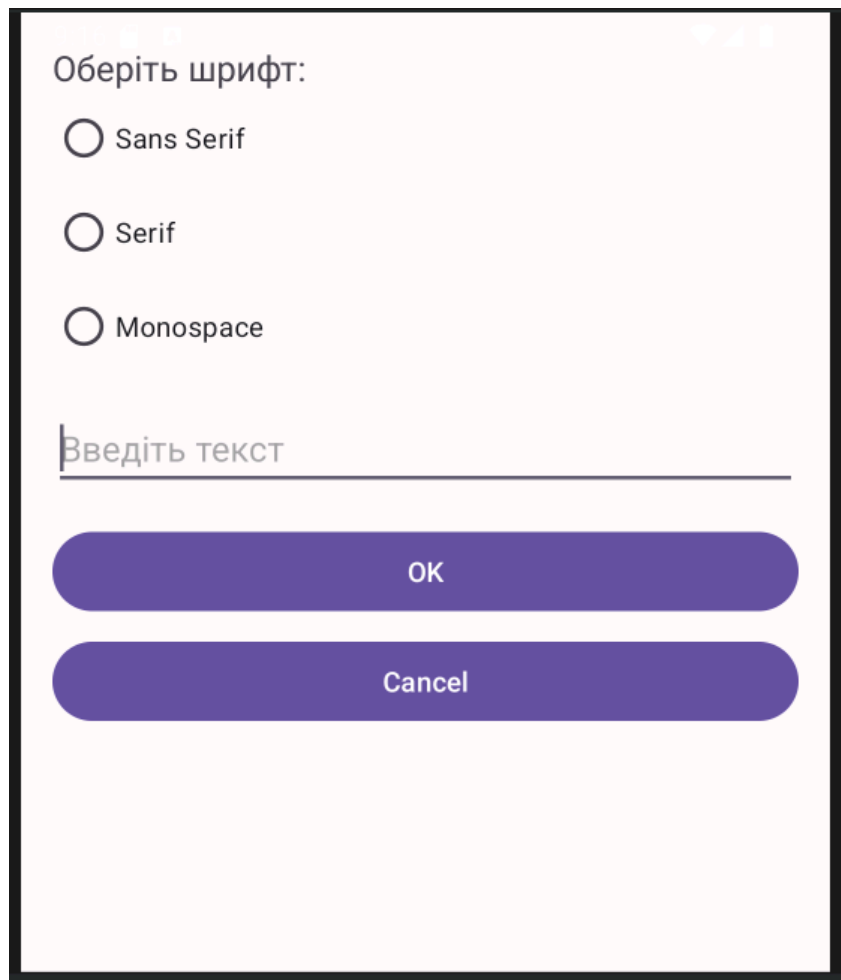
Написати програму під платформу Андроїд, яка має інтерфейс для введення або/та вибору даних згідно варіанту (таблиця) і відображає результат взаємодії з цим інтерфейсом у деяке текстове поле цього інтерфейсу. Передбачити наступне: якщо не всі дані введені або обрані, а користувач натискає кнопку для отримання результату, то відобразити вікно, що спливає, з повідомленням завершити введення всіх даних.

### **Варіант №3**

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

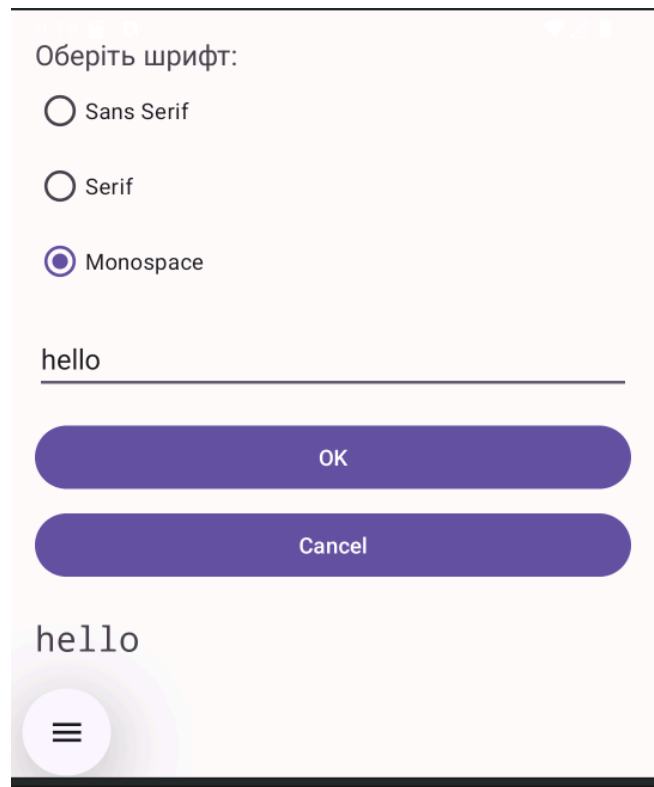
### **1. Реалізація**

Розробка застосунку здійснювалася в середовищі Android Studio з використанням мови програмування Kotlin. Інтерфейс користувача реалізовано у файлі `activity_main.xml` із застосуванням стандартних елементів Android. Основним контейнером було використано `LinearLayout` з вертикальною орієнтацією, що дозволило розмістити елементи інтерфейсу послідовно один під одним. Для вибору типу шрифту застосовано компонент `RadioGroup`, який об'єднує декілька `RadioButton` та забезпечує можливість вибору лише одного варіанта одночасно. Для введення текстової інформації використано елемент `EditText`. Кнопки керування реалізовано за допомогою компонентів `Button` з написами «OK» та «Cancel». Відображення результату здійснюється у `TextView`.

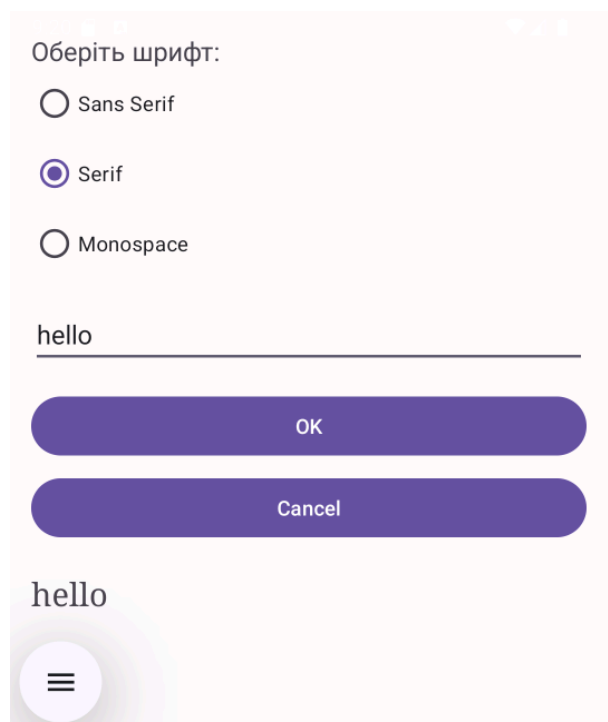


*Рисунок 1 – Головне вікно розробленого застосунку*

Програмна логіка реалізована у файлі MainActivity.kt. У методі onCreate було встановлено обробники натискань для кнопок. При натисканні кнопки «OK» програма отримує введений користувачем текст та визначає, чи вибрано один із варіантів шрифту. Якщо текстове поле порожнє або радіо-кнопка не обрана, застосунок виводить спливаюче повідомлення за допомогою класу Toast з текстом про необхідність завершити введення всіх даних. Якщо ж введення виконано коректно, програма визначає обраний тип шрифту та застосовує його до елемента TextView за допомогою класу Typeface, після чого відображає введений рядок. Кнопка «Cancel» реалізує очищення поля введення, текстового поля результату та скидання вибору радіо-кнопки.



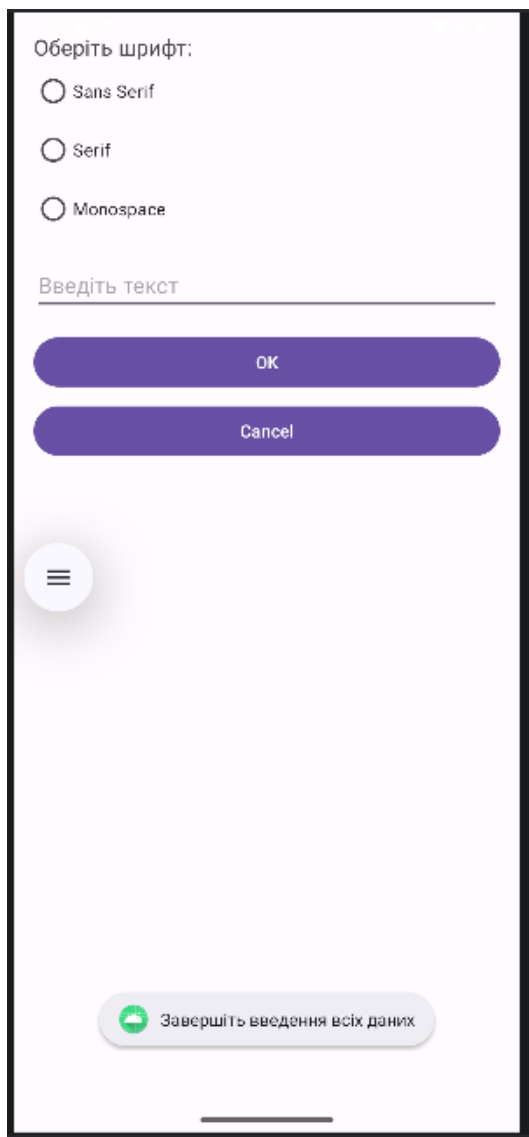
*Рисунок 2 – Відображення введеного тексту зі шрифтом Monospace*



*Рисунок 3 – Відображення введеного тексту з іншим обраним шрифтом*

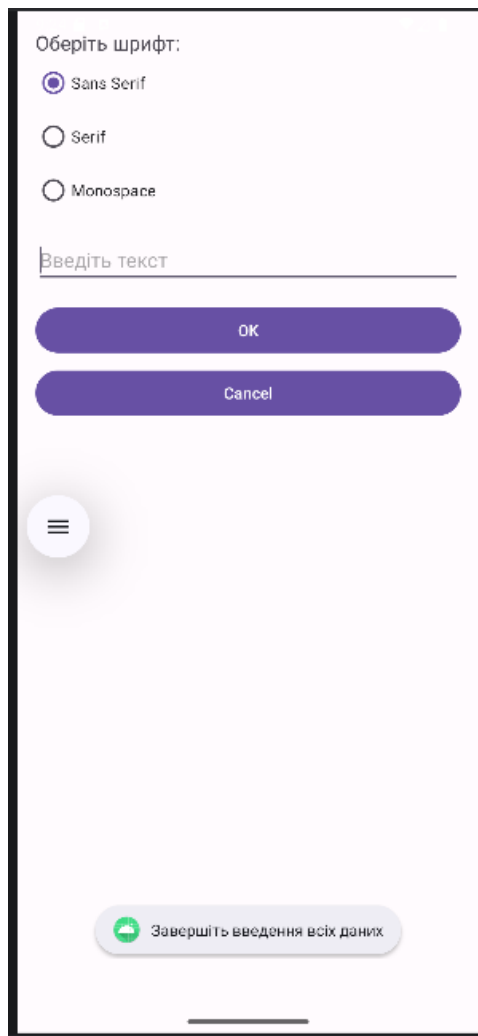
## 2. Тестування

У процесі тестування було перевірено декілька сценаріїв роботи програми. При натисканні кнопки «ОК» без введення тексту та без вибору шрифту коректно відображалося повідомлення про необхідність заповнення всіх даних.

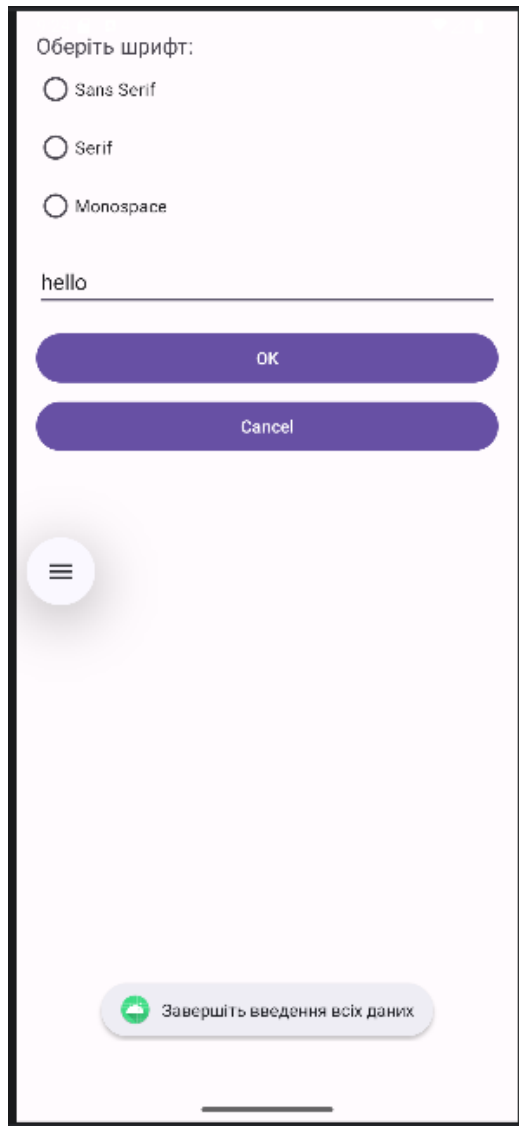


*Рисунок 4 – Відображення повідомлення про необхідність введення всіх даних*

Аналогічний результат спостерігався у випадку часткового введення, коли було введено лише текст або обрано лише шрифт. Після введення тексту та вибору одного з варіантів шрифту результат коректно відображався у нижньому текстовому полі з відповідним форматуванням.

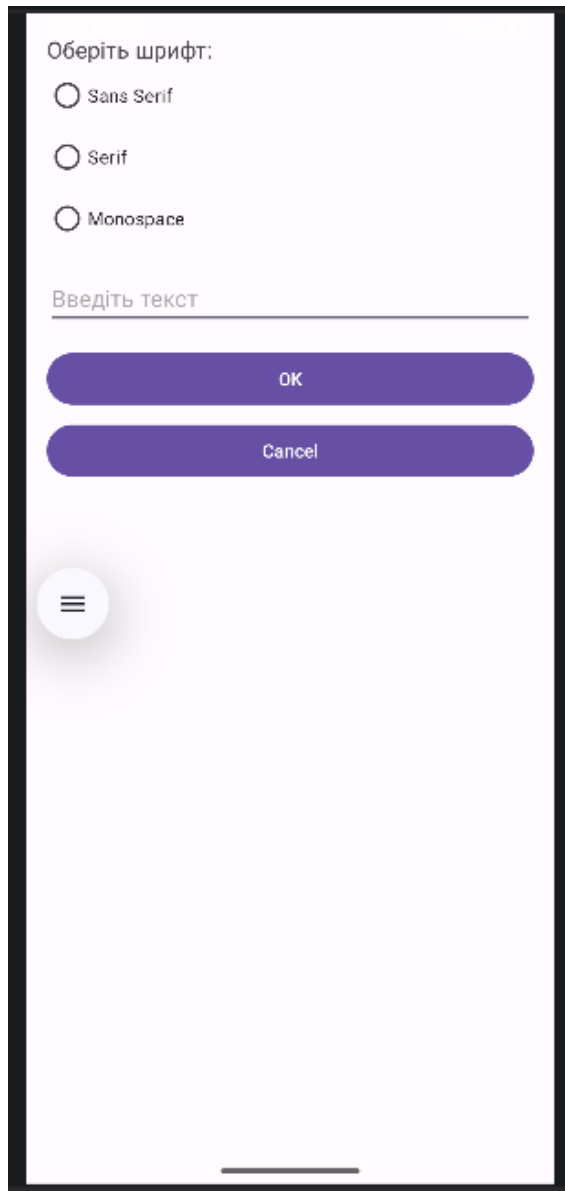


*Рисунок 5 – Відображення повідомлення про необхідність введення всіх даних (текст відсутній)*



*Рисунок 6 – Відображення повідомлення про необхідність введення всіх даних (шрифт відсутній)*

Натискання кнопки «Cancel» повністю очищало всі дані інтерфейсу.



*Рисунок 7 – Очищення введених даних після натискання кнопки Cancel*

## **Контрольні питання**

### **1. Архітектура застосунку під платформу Android**

Архітектура застосунку під платформу Android базується на компонентному підході. Кожен застосунок складається з окремих незалежних компонентів, які взаємодіють між собою через механізм намірів (Intent). Android використовує багаторівневу архітектуру, що включає рівень застосунків, фреймворк застосунків, бібліотеки та ядро Linux. На рівні фреймворку надаються засоби керування активностями,

службами, ресурсами, повідомленнями та іншими системними можливостями.

## **2. Загальний огляд компонентів застосунку під Android**

Основними компонентами Android-застосунку є Activity, Service, Broadcast Receiver та Content Provider. Activity відповідає за взаємодію з користувачем та відображення інтерфейсу. Service працює у фоновому режимі та виконує тривалі операції без взаємодії з користувачем. Broadcast Receiver реагує на системні або прикладні події, наприклад зміну стану мережі. Content Provider забезпечує доступ до структурованих даних та дозволяє обмін даними між застосунками.

## **3. Життєвий цикл компоненту «Діяльність»**

Життєвий цикл Activity визначається системою Android і складається з декількох основних методів: onCreate(), onStart(), onResume(), onPause(), onStop() та onDestroy(). Метод onCreate() викликається при створенні Activity і використовується для ініціалізації інтерфейсу та змінних. onStart() означає, що Activity стає видимою для користувача. onResume() вказує, що Activity переходить у активний стан і готова до взаємодії. onPause() викликається, коли Activity частково перекривається іншою. onStop() означає, що Activity більше не видима. onDestroy() викликається перед остаточним знищенням об'єкта.

## **4. Життєвий цикл компоненту «Служба»**

Служба призначена для виконання фонових операцій і має власний життєвий цикл. Основними методами є onCreate(), onStartCommand() та onDestroy(). Метод onCreate() викликається при створенні служби, onStartCommand() — коли служба запускається, а onDestroy() — перед її завершенням. Існують два типи служб: запущені (Started Service) та зв'язані (Bound Service). Запущена служба працює незалежно, доки її не буде зупинено, а зв'язана служба існує лише поки до неї підключені інші компоненти.

## **5. Опис процесів платформи Android**

Кожен застосунок Android працює у власному ізольованому процесі та має окрему віртуальну машину. Система керує процесами автоматично,



враховуючи їхню важливість. Існують процеси переднього плану, видимі процеси, фонові та кешовані процеси. Якщо системі бракує пам'яті, вона може завершити менш пріоритетні процеси.

## **6. Яким чином активуються компоненти застосунку**

Компоненти Android-застосунку активуються за допомогою механізму Intent. Intent — це об'єкт, який описує намір виконати певну дію. Він може використовуватися для запуску Activity, Service або надсилання широкомовного повідомлення. Існують явні (explicit) та неявні (implicit) наміри. Явний намір прямо вказує компонент, який потрібно запустити, а неявний — описує дію, і система самостійно визначає відповідний компонент.

## **7. Призначення файлу маніфесту та його структура**

Файл AndroidManifest.xml є обов'язковим для кожного застосунку та містить основну інформацію про нього. У маніфесті визначаються всі компоненти застосунку, права доступу, мінімальна версія SDK, а також інші параметри конфігурації. Структурно файл складається з кореневого елемента <manifest>, всередині якого містяться елементи <application>, <activity>, <service> та інші. Маніфест використовується системою Android для визначення структури застосунку та правил його роботи.

## **8. Поняття ресурсу та яким чином визначаються ресурси**

Ресурси в Android — це зовнішні файли, які використовуються застосунком і не містять програмного коду. До ресурсів належать макети інтерфейсу, рядкові значення, зображення, стилі, кольори та інші елементи. Вони зберігаються у папці res та організовуються за типами. Доступ до ресурсів здійснюється через клас R, який автоматично генерується системою. Використання ресурсів дозволяє розділити логіку програми та її представлення, а також спрощує локалізацію та підтримку різних конфігурацій пристроїв.

### Висновки:

У ході виконання лабораторної роботи було засвоєно принципи побудови інтерфейсу Android-застосунку, обробки подій натискання кнопок, перевірки введених даних та зміни властивостей елементів інтерфейсу програмним шляхом. Отримано практичні навички роботи з XML-розміткою та мовою Kotlin для реалізації логіки взаємодії користувача з програмою. Створений застосунок повністю відповідає поставленому завданню та коректно виконує всі передбачені функції.

Репозиторій GitHub: [https://github.com/ProfessionalA1k0/Lab1\\_Android](https://github.com/ProfessionalA1k0/Lab1_Android)

### Код програми:

MainActivity.kt:

```
package com.example.lab1

import android.graphics.Typeface
import android.os.Bundle
import android.widget.*
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val rgFont = findViewById<RadioGroup>(R.id.rgFont)
        val etInput = findViewById<EditText>(R.id.etInput)
        val tvResult = findViewById<TextView>(R.id.tvResult)
        val btnOk = findViewById<Button>(R.id.btnOk)
        val btnCancel = findViewById<Button>(R.id.btnCancel)

        btnOk.setOnClickListener {
            val text = etInput.text.toString()
            val selectedId = rgFont.checkedRadioButtonId

            if (text.isBlank() || selectedId == -1) {
                Toast.makeText(this, "Завершіть введення всіх даних", Toast.LENGTH_SHORT).show()
                return@setOnClickListener
            }

            val typeface = when (selectedId) {
```

```

        R.id.rbSans -> Typeface.SANS_SERIF
        R.id.rbSerif -> Typeface.SERIF
        R.id.rbMono -> Typeface.MONOSPACE
        else -> Typeface.DEFAULT
    }

    tvResult.typeface = typeface
    tvResult.text = text
}

btnCancel.setOnClickListener {
    etInput.text.clear()
    tvResult.text = ""
    rgFont.clearCheck()
}
}
}

```

activity\_main.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:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Оберить шрифт:"
        android:textSize="18sp"/>

    <RadioGroup
        android:id="@+id/rgFont"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <RadioButton
            android:id="@+id/rbSans"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Sans Serif"/>

        <RadioButton
            android:id="@+id/rbSerif"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

```

```
        android:text="Serif"/>

        <RadioButton
            android:id="@+id/rbMono"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Monospace"/>
    </RadioGroup>

    <EditText
        android:id="@+id/etInput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Введите текст"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/btnOk"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="OK"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/btnCancel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Cancel"
        android:layout_marginTop="8dp"/>

    <TextView
        android:id="@+id/tvResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="22sp"
        android:layout_marginTop="20dp"/>
</LinearLayout>
```