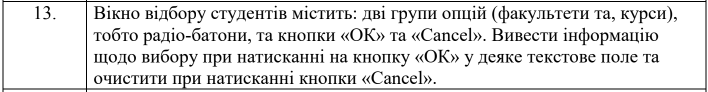
**Лаборатона робота 1 Окара Сергій ІК-21**

ДОСЛІДЖЕННЯ РОБОТИ З ЕЛЕМЕНТАМИ КЕРУВАННЯ

**Мета роботи:** дослідити створення простого застосунку під платформу Андроїдта набути практичні навички з використання елементів керування інтерфейсу, мовпрограмування Java чи Kotlin.

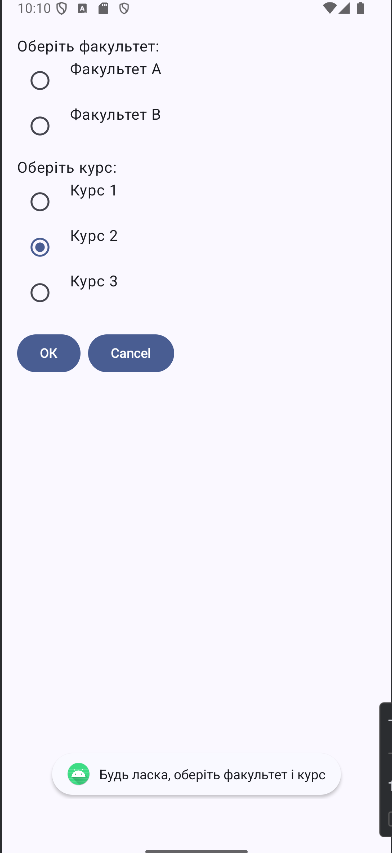
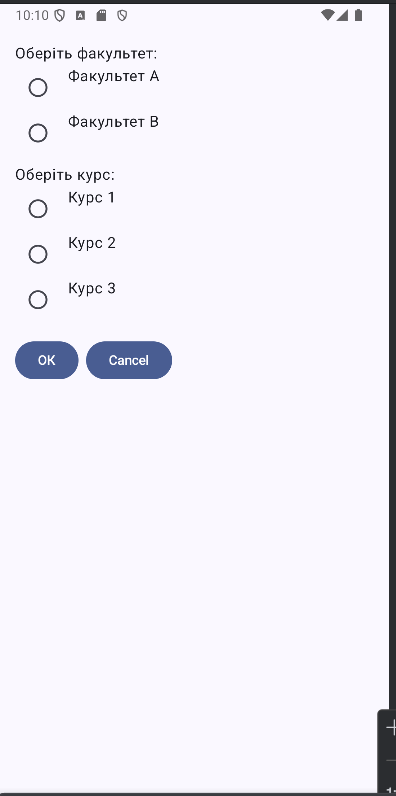
Варіант 13



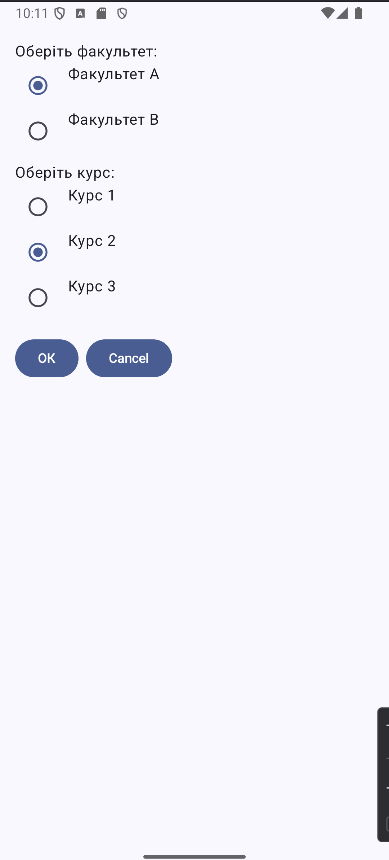
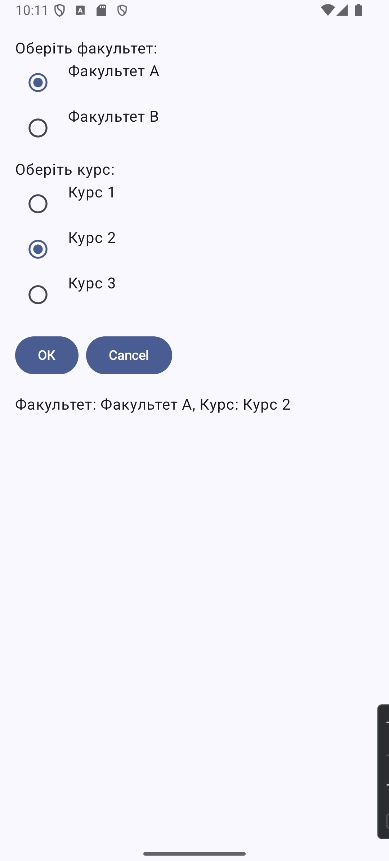
Вікно відбору студентів реалізовано за допомогою Jetpack Compose. Інтерфейс містить дві групи радіо-кнопок (для вибору факультету та курсу), а також кнопки "ОК" та "Cancel".

* При натисканні кнопки "ОК", якщо користувач обрав обидва параметри, результат виводиться у текстове поле.
* Якщо жоден або один із параметрів не вибрано, відображається повідомлення Toast з відповідним нагадуванням.
* При натисканні кнопки "Cancel", всі вибори очищуються.

Використано MutableState для збереження стану вибору.



Якщо обрати тільки один елемент програма, запропонує обрати всі відповіді, та не буде виводити текст доки не обрати два елементи.



Натискання на кнопку «ОК» виводить обраний текст, а натискання на кнопку «Саncel» видаляє його.

**Програмний код:**

package com.example.myapplication  
  
import android.os.Bundle  
import android.widget.Toast  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.activity.enableEdgeToEdge  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.platform.LocalContext  
import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import com.example.myapplication.ui.theme.MyApplicationTheme  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 enableEdgeToEdge()  
 setContent {  
 MyApplicationTheme {  
 Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->  
 SelectionScreen(modifier = Modifier.padding(innerPadding))  
 }  
 }  
 }  
 }  
}  
  
@Composable  
fun SelectionScreen(modifier: Modifier = Modifier) {  
 val context = LocalContext.current  
 var selectedFaculty by remember { mutableStateOf("") }  
 var selectedCourse by remember { mutableStateOf("") }  
 var resultText by remember { mutableStateOf("") }  
  
 Column(modifier = modifier.padding(16.dp)) {  
 Text(text = "Оберіть факультет:")  
 RadioGroup(options = listOf("Факультет A", "Факультет B")) { selectedFaculty = it }  
  
 Spacer(modifier = Modifier.height(8.dp))  
  
 Text(text = "Оберіть курс:")  
 RadioGroup(options = listOf("Курс 1", "Курс 2", "Курс 3")) { selectedCourse = it }  
  
 Spacer(modifier = Modifier.height(16.dp))  
  
 Row {  
 Button(onClick = {  
 if (selectedFaculty.isNotEmpty() && selectedCourse.isNotEmpty()) {  
 resultText = "Факультет: $selectedFaculty, Курс: $selectedCourse"  
 } else {  
 Toast.makeText(context, "Будь ласка, оберіть факультет і курс", Toast.LENGTH\_SHORT).show()  
 }  
 }) {  
 Text("ОК")  
 }  
 Spacer(modifier = Modifier.width(8.dp))  
 Button(onClick = { selectedFaculty = ""; selectedCourse = ""; resultText = "" }) {  
 Text("Cancel")  
 }  
 }  
 Spacer(modifier = Modifier.height(16.dp))  
 Text(text = resultText)  
 }  
}  
  
@Composable  
fun RadioGroup(options: List<String>, onOptionSelected: (String) -> Unit) {  
 var selectedOption by remember { mutableStateOf("") }  
  
 Column {  
 options.forEach { option ->  
 Row {  
 RadioButton(  
 selected = (option == selectedOption),  
 onClick = {  
 selectedOption = option  
 onOptionSelected(option)  
 }  
 )  
 Text(text = option, modifier = Modifier.padding(start = 8.dp))  
 }  
 }  
 }  
}  
  
@Preview(showBackground = true)  
@Composable  
fun SelectionScreenPreview() {  
 MyApplicationTheme {  
 SelectionScreen()  
 }  
}

**Контрольні питання**

1. Архітектура застосунку під платформу Андроїд

Android-застосунки використовують багатошарову архітектуру:

* UI-рівень – реалізований через Jetpack Compose.
* Логіка застосунку – керує виборами користувача.
* Рівень даних – у цій програмі стан зберігається у пам'яті (MutableState).
* Android Framework – забезпечує управління активностями, сервісами тощо.
* Linux Kernel – працює з апаратними ресурсами.

2. Загальний огляд компонентів застосунку під Андроїд

Основні компоненти Android-застосунку:

* Activity – використовується для відображення інтерфейсу.
* Service – виконує фонові операції (у цьому додатку не використовується).
* Broadcast Receiver – обробляє системні події.
* Content Provider – забезпечує обмін даними між додатками.

3. Життєвий цикл компоненту «Діяльність»

Життєвий цикл Activity включає такі етапи:

1. onCreate() – ініціалізація.
2. onStart() – підготовка до відображення.
3. onResume() – активний стан.
4. onPause() – часткове припинення.
5. onStop() – повна зупинка.
6. onDestroy() – знищення об'єкта.
7. onRestart() – відновлення після зупинки.

4. Життєвий цикл компоненту «Служба»

Служба може бути запущеною (Started Service) або прив'язаною (Bound Service). Основні методи:

1. onCreate() – створення сервісу.
2. onStartCommand() – початок виконання завдання.
3. onBind() – прив'язка до сервісу.
4. onUnbind() – від'єднання клієнта.
5. onDestroy() – завершення роботи.

5. Опис процесів платформи Андроїд

Android керує процесами за рівнями пріоритету:

1. Foreground Process – активна Activity або Service.
2. Visible Process – UI видимий, але не у фокусі.
3. Service Process – фоновий сервіс.
4. Cached Process – неактивний додаток.

6. Яким чином активуються компоненти застосунку

Компоненти Android активуються через інтенти (Intent):

* Явний (Explicit Intent) – викликає конкретний компонент (startActivity(Intent(this, MainActivity::class.java))).
* Неявний (Implicit Intent) – використовується для дій (ACTION\_VIEW).
* Broadcast Intent – для надсилання подій.
* Pending Intent – для запланованих дій.

7. Призначення файлу маніфесту та його структура

Файл AndroidManifest.xml містить інформацію про застосунок, наприклад:

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.app">

<application>

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

Основні розділи:

* <application> – загальні налаштування.
* <activity> – список активностей.
* <service> – фонові служби.
* <receiver> – Broadcast Receivers.
* <provider> – Content Providers.
* <uses-permission> – дозволи для доступу до системних ресурсів.

8. Поняття ресурсу та яким чином визначаються ресурси

Ресурси – це статичні дані, які не є частиною коду:

* Макети: res/layout/activity\_main.xml
* Рядки: res/values/strings.xml
* Зображення: res/drawable/
* Кольори: res/values/colors.xml
* Стилі: res/values/styles.xml

Використання у коді:

val title = getString(R.string.app\_name)

val color = ContextCompat.getColor(this, R.color.black)

Висновки

Реалізовано вікно відбору студентів та загальні концепції платформи Android, включаючи її архітектуру, життєвий цикл компонентів. Додаток створений з використанням Jetpack Compose, що забезпечує зручну та сучасну розробку UI.