

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені
ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до лабораторної роботи №5

з курсу

«Архітектура мобільних операційних систем і пристроїв»

*Студентки 4 курсу
групи ПП-41*

спеціальності 122 «Комп'ютерні науки»

ОП «Прикладне програмування»

Штось Софії Максимівни

Викладач:

Краснощок В.М.

Київ – 2023

Тема роботи: Робота із віджетами та керуючими елементами.

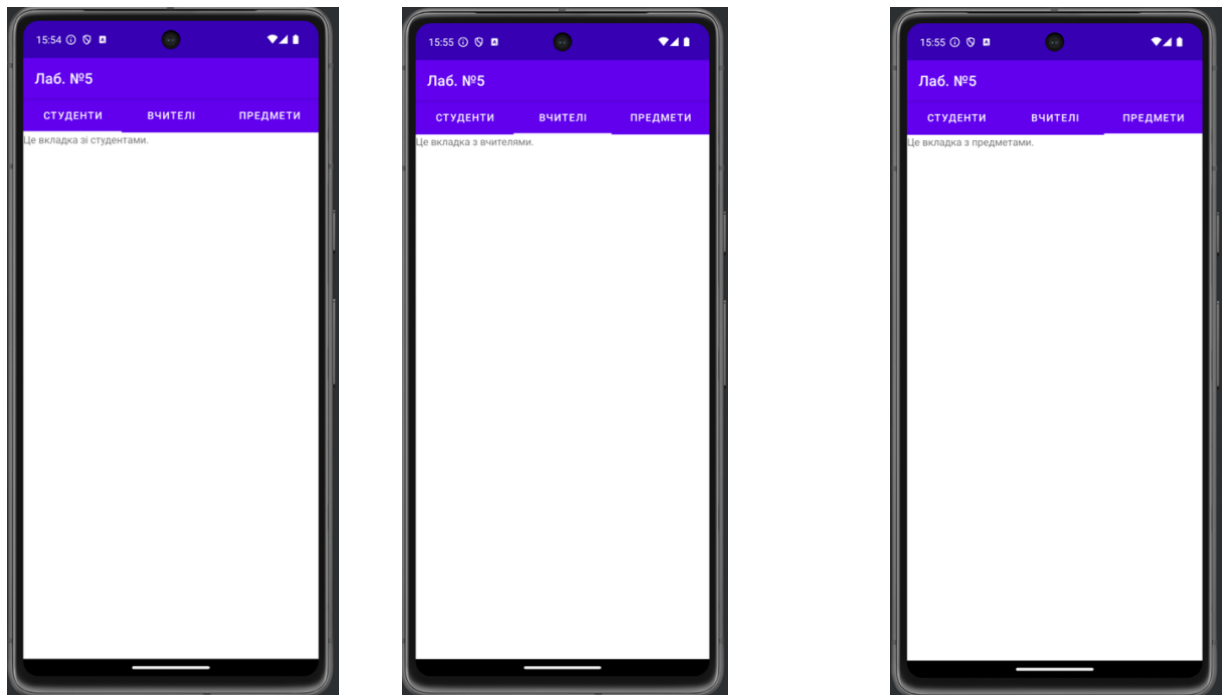
Мета роботи: Ознайомитись з використанням віджетів TabWidget, WebView, ListView. Навчитися використовувати в інтерфейсі користувача різні керуючі елементи: кнопки із зображеннями, радіокнопки, чекбокси та ін.

Завдання для виконання (варіант 23)

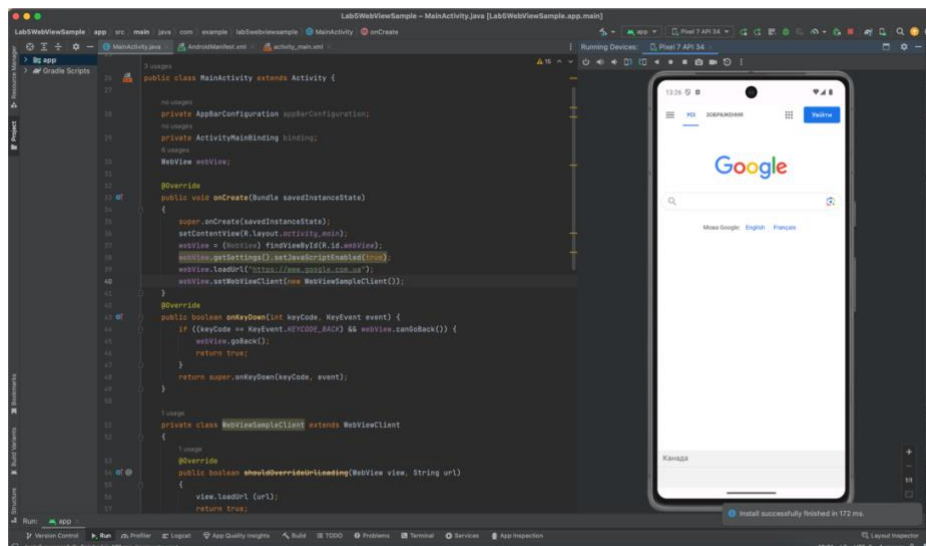
Виконайте завдань до виконання у відповідності до теоретичних відомостей поданих вище.

Хід роботи

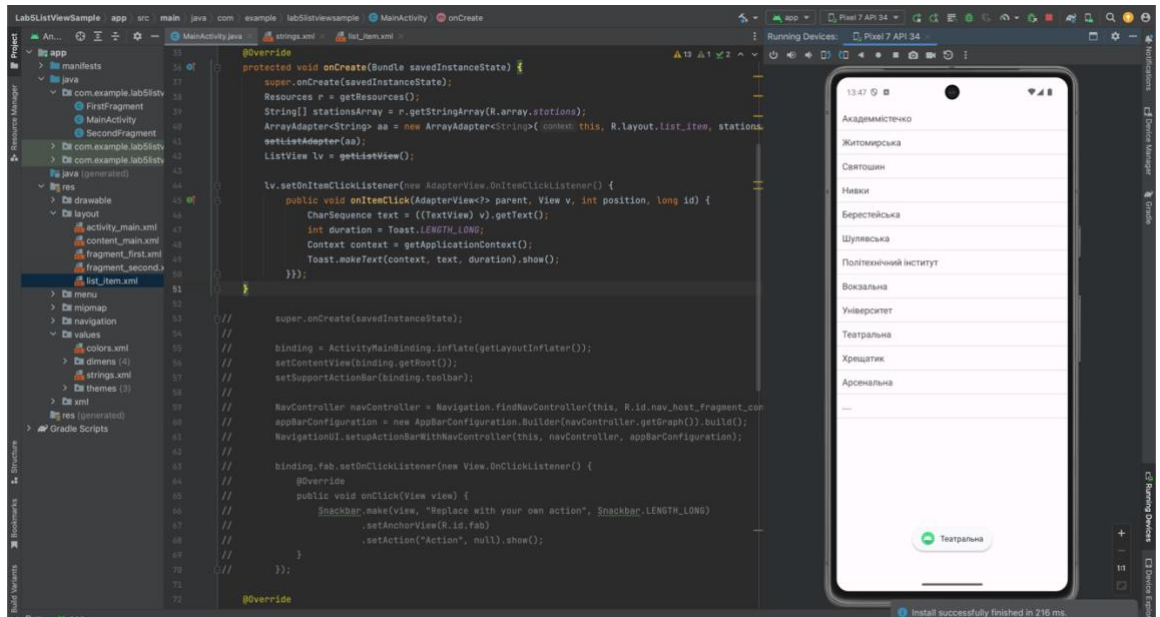
Завдання 1. «Робота з вкладками»



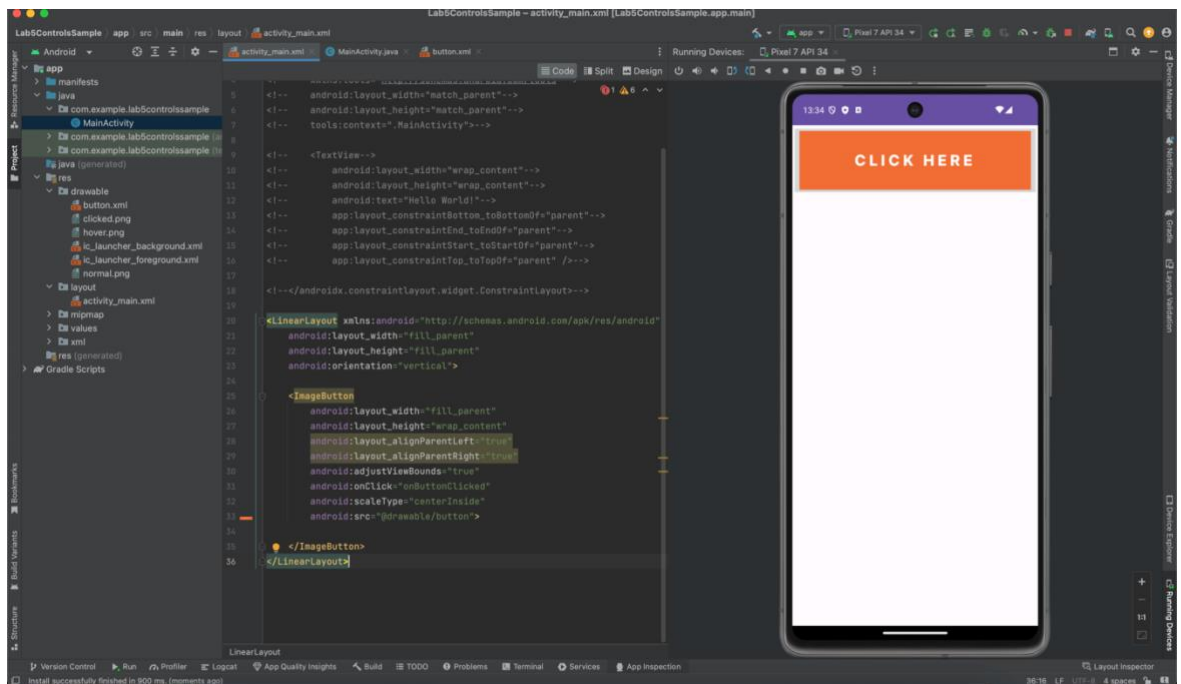
Завдання 2. «Використання WebView»

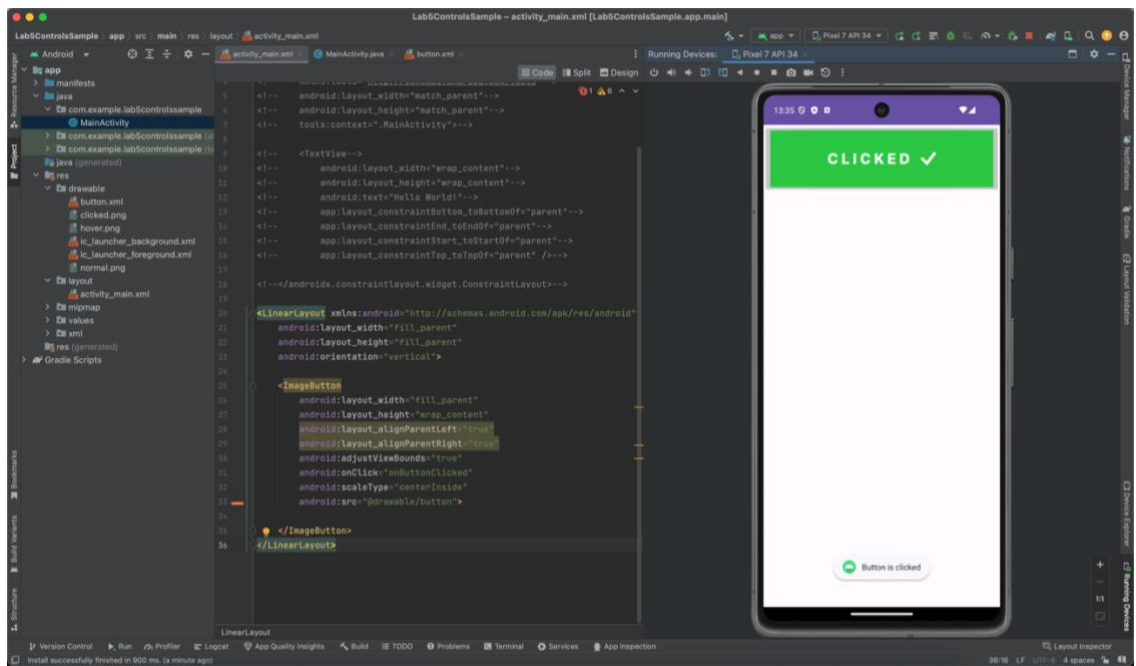


Завдання 3. «Використання ListView»

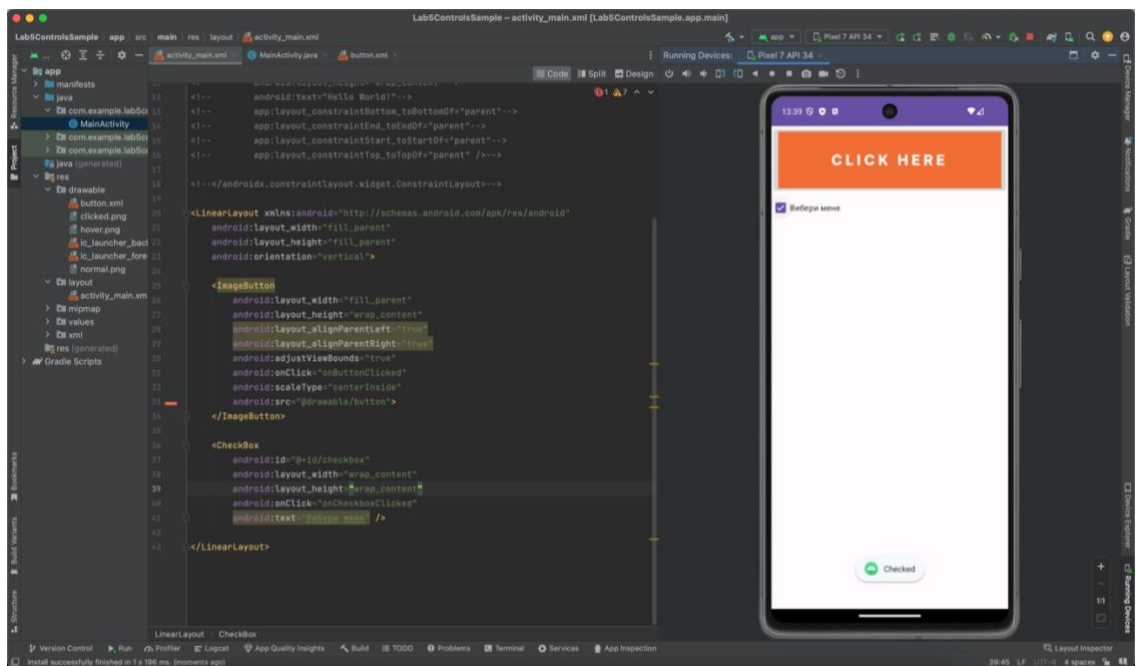


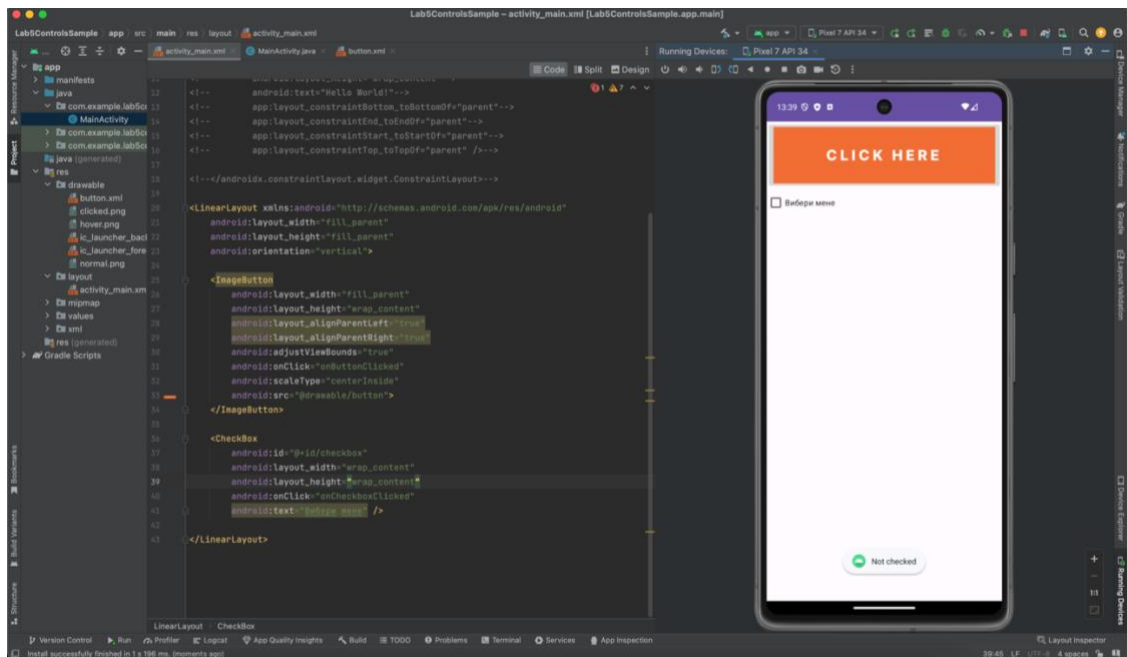
Завдання 4. «Використання керуючих елементів в інтерфейсі»



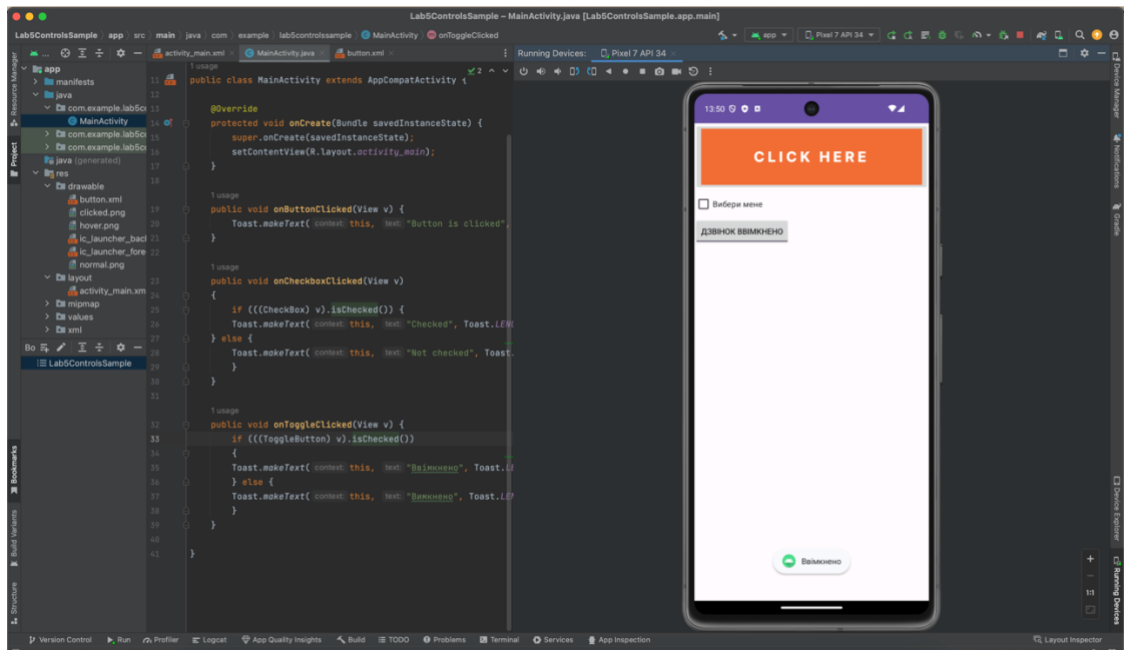


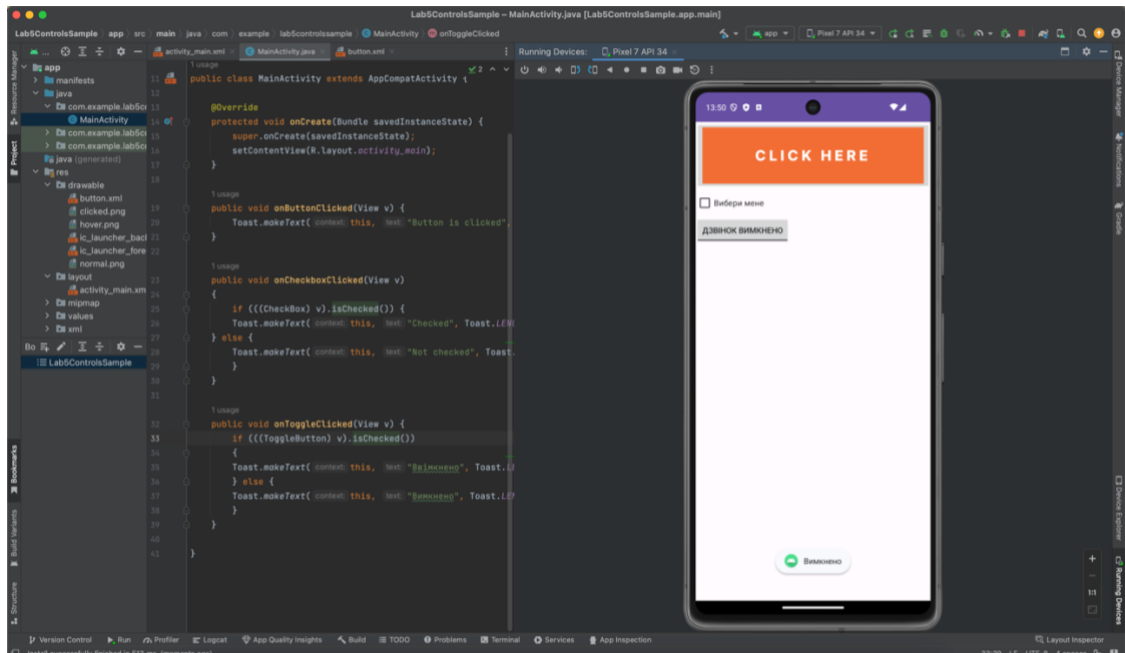
Завдання 5. «Використання віджета CheckBox»



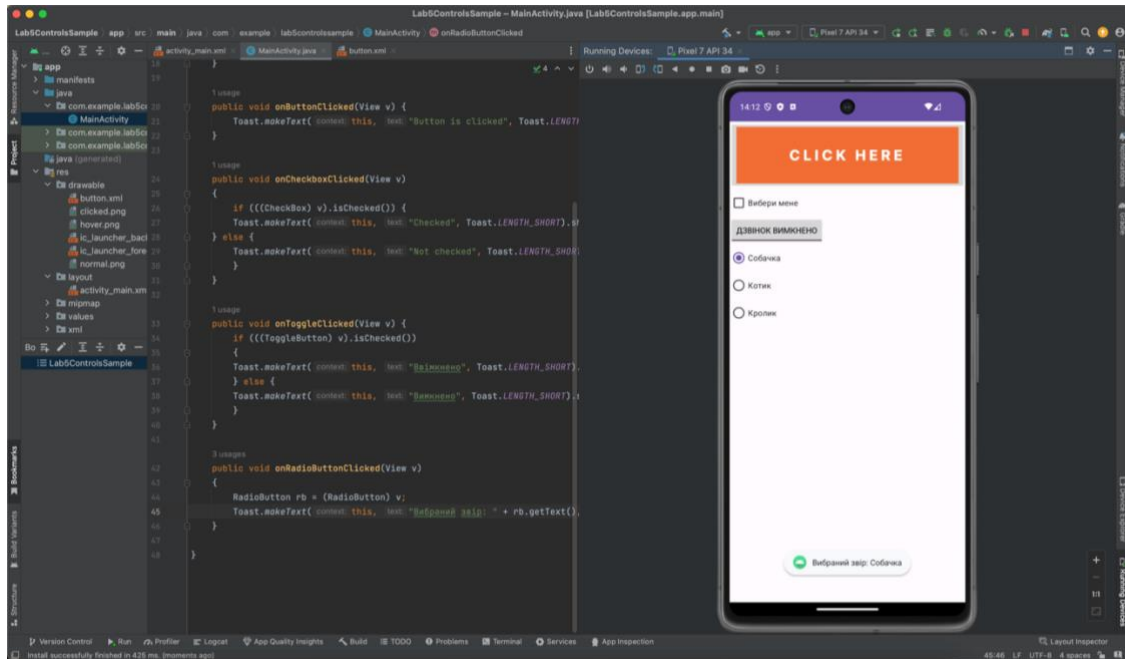


Завдання 6. «Використання віджета ToggleButton»





Завдання 7. «Використання віджета RadioButton»



Завдання 8. «Використання віджета EditText»

Висновок: Отже, у ході цієї лабораторної роботи було проведено роботу з віджетами та керуючими елементами в Android. Було проведено ознайомлення з такими елементами, як TabWidget, WebView, ListView, кнопки із зображеннями, радіокнопки, чекбокси та ін.

Контрольні запитання

1. У чому відмінність між методом *setOnClickListener()* і методом *setOnClickListener()* елемента *ListView*?

setOnClickListener() використовують, коли елемент клацнуто (один вибір), тоді як *setOnClickListener()* використовують, коли елемент вибрано (використовується для багаторазового вибору).

2. Який механізм виступає в ролі «сполучної ланки» між джерелом даних і елементом *ListView*?

"Сполучною ланкою" між джерелом даних і елементом *ListView* є адаптер. Адаптер виступає як посередник, що взаємодіє з джерелом даних (таким як масив чи база даних) і передає ці дані до елементів *ListView* для їх відображення. Адаптер також відповідає за управління виглядом кожного елемента списку. Він надає необхідний міст між структурою даних і відображенням на екрані.

3. Який елемент-контейнер найкраще підходить для вирівнювання дочірніх елементів *View* щодо батьківського елемента?

Для вирівнювання дочірніх елементів *View* щодо батьківського елемента найкраще використовувати *RelativeLayout*. У *RelativeLayout* можна визначати, як кожен дочірній елемент повинен бути вирівняний відносно інших дочірніх елементів чи щодо батьківського контейнера за допомогою різних правил розташування. Це надає більше гнучкості у вирівнюванні та розміщенні елементів на екрані.

4. Особливості використання *RelativeLayout*.

RelativeLayout — це View-група, яка відображає дочірні види у relative позиціях. Позицію кожного View можна вказати відносно однорідних елементів (наприклад, ліворуч або під іншим поданням) або по відношенню до батьківської області RelativeLayout (наприклад, вирівняти знизу, зліва чи по центру).

RelativeLayout — це дуже потужна утиліта для розробки інтерфейсу користувача, оскільки вона може усунути вкладені групи переглядів і зберегти ієрархію макета рівною, що покращує продуктивність. Якщо використовуються кілька вкладених груп LinearLayout, їх можна замінити однією RelativeLayout.

5. *Обробка подій життєвого циклу додатку*

Обробка подій життєвого циклу додатку в Android дуже важлива для правильної реалізації функціоналу. Основні методи життєвого циклу додатку включають:

`onCreate()`: Викликається при створенні активності або служби. Тут ви можете ініціалізувати ресурси, які потрібні для вашого додатку.

`onStart()`: Викликається, коли активність стає видимою для користувача, але ще не готова до взаємодії.

`onResume()`: Викликається, коли активність стає активною і готовою до взаємодії з користувачем.

`onPause()`: Викликається, коли інша активність стає активною, а поточна активність переходить в фоновий режим. Тут важливо зберегти стан додатку.

`onStop()`: Викликається, коли активність більше не видима для користувача. Тут можна виконати операції, які необхідно завершити, коли активність не використовується.

`onDestroy()`: Викликається при завершенні життєвого циклу активності або служби. Тут ви можете звільнити ресурси та виконати завершальні операції.

`onRestart()`: Викликається, коли активність рестартується після зупинки.

