

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені
ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до лабораторної роботи №4

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 анімацією та макетами в ОС Android.

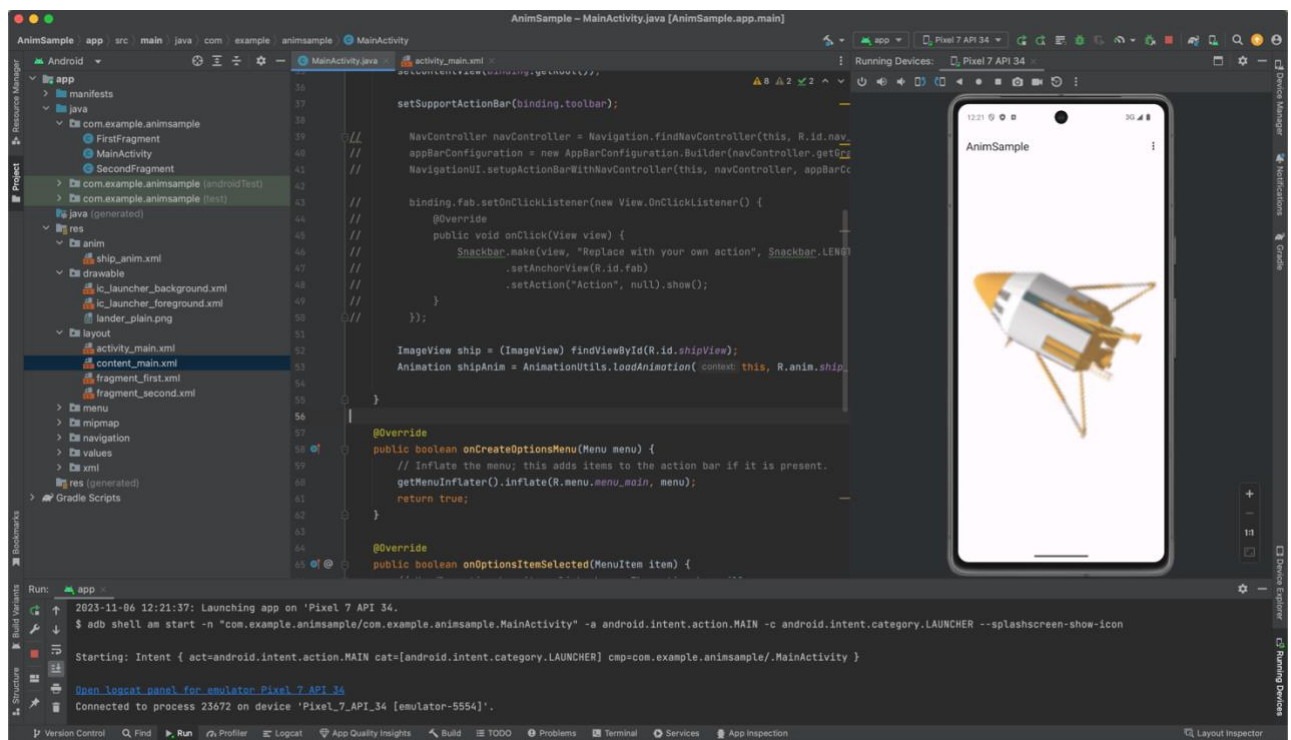
Мета роботи: Отримати досвід роботи із анімацією та ознайомитись із основними макетами для побудови користувацького інтерфейсу додатків під ОС Android.

Завдання для виконання (варіант 23)

Використовуючи програму створену в 3й практичній роботі додайте до неї необхідні елементи для виконання в ній зміни у відповідності до завдань які розглянуті у пунктах теоретичних відомостей поданих вище..

Хід роботи

Завдання 1. «Робота з анімацією»



Завдання 2. «Використання *LinearLayout*»

activity_main.xml:

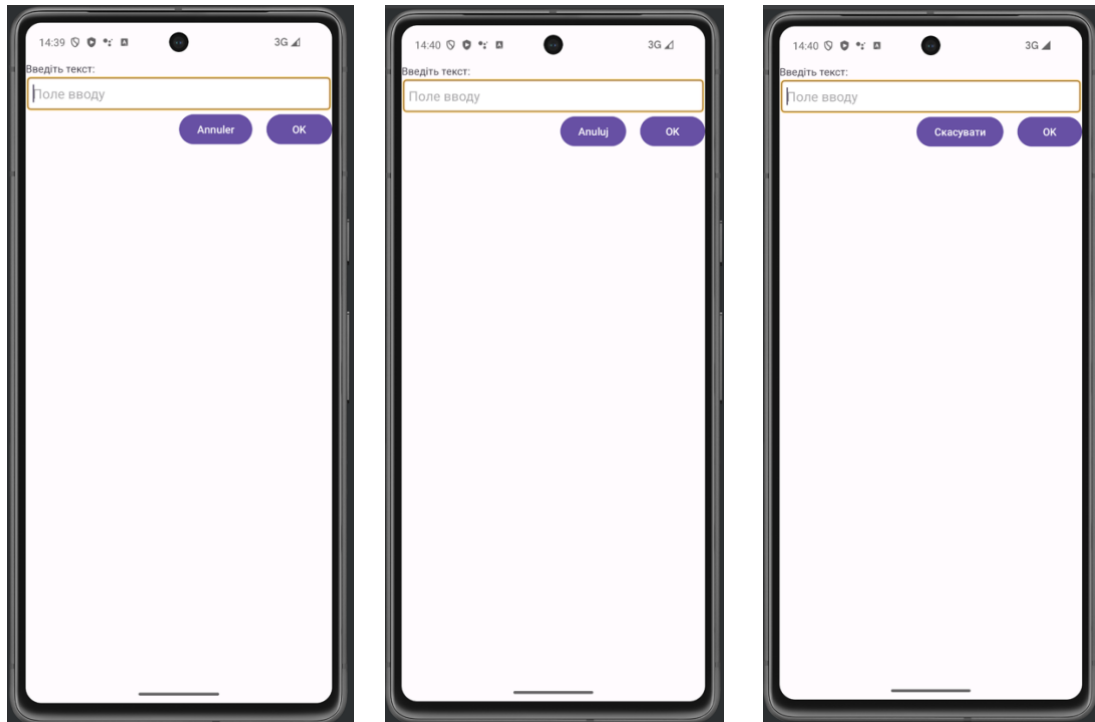
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:background="@color/black"
    android:orientation="vertical">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="@string/helloworld"
        android:textColor="@color/white" />

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:contentDescription="@string/app_name"
        android:src="@drawable/background_image" />
</LinearLayout>
```



Завдання 3. «Використання *RelativeLayout*»



Висновок: Отже, у ході цієї лабораторної роботи було проведено роботу з анімацією та макетами в ОС Android. Було отримано навички роботи із анімацією та ознайомлення із основними макетами для побудови користувацького інтерфейсу додатків під ОС Android.

Контрольні запитання

1. *Наскільки добре платформа Android справляється з анімацією?*

Анімація додає візуальні підказки, які сповіщають користувачів про те, що відбувається у програмі. Вони особливо корисні, коли інтерфейс користувача змінює стан, наприклад, коли завантажується новий вміст або стають доступними нові дії. Android включає різні API анімації залежно від того, який тип анімації потрібен.

2. *Чи існує спосіб зупинки відтворення анімації?*

Щоб зупинити анімацію, можна викликати метод `cancel()`.

3. *Які типи операцій підтримуються анімаціями руху?*

- Зміна положення (Translation): Переміщення об'єкта вздовж вісей X, Y.
- Зміна розміру (Scale): Збільшення або зменшення розмірів об'єкта.
- Обертання (Rotation): Обертання об'єкта навколо своєї вісі.

- Прозорість (Alpha): Зміна прозорості об'єкта від повністю прозорого до повністю видимого.

4. *Чи може бути використаний елемент-контейнер `LinearLayout` для розміщення всіх дочірніх елементів-уявлень `View` один за одним (по вертикалі)?*

Властивість `orientation` `LinearLayout` дозволяє вибрати, чи контейнер відображатиме свої дочірні елементи по горизонталі чи по вертикалі.

Наприклад, якщо потрібно розмістити всі дочірні елементи по вертикалі, можна встановити `android:orientation="vertical"` в елементі `LinearLayout` в розмітці XML.

5. *Назвіть віджети, які використовуються для створення користувацького інтерфейсу.*

Клас `View` є базовим класом для всіх візуальних елементів UI (елементів управління (`Control`) і віджетів (`Widget`)). Всі ці елементи, в тому числі і розмітка (`Layout`), є розширеннями класу `View`.

Android надає розробнику можливість використання набору готових елементів користувацького інтерфейсу, що прописані у класі `View`:

- `TextView`. Стандартний елемент, призначений для виведення тексту. Підтримує багаторядкове відображення, форматування і автоматичний перенос.
- `EditText`. Редаговане поле для введення тексту. Підтримує багаторядкове введення, перенесення слів на новий рядок і текст підказки.
- `ListView`. Група уявлень (`ViewGroup`), яка формує вертикальний список елементів, відображаючи їх у вигляді рядків всередині списку. Найпростіший об'єкт `ListView` використовує `TextView` для виводу на екран значень `toString()`, що належать елементом масиву.
- `Spinner`. Складовий елемент, що відображає `TextView` у поєднанні з відповідним `ListView`, який дозволяє вибрати елемент списку для відображення в текстовому рядку. Сам рядок складається з об'єкта `TextView` і кнопки при натисканні на яку починається діалог вибору. Зовні цей елемент нагадує тег `<SELECT>` в HTML.
- `Button`. Стандартна кнопка, яку можна натискати.
- `Checkbox`. Кнопка, що має два стани. Представлена у вигляді прапорця.

- **RadioButton.** «Радіокнопка», яка дозволяє вибрати тільки один з декількох варіантів.

- **ViewFlipper.** Група уявлень (ViewGroup), що дозволяє визначити набір елементів і горизонтальний рядок, в якому може виводитися тільки одне View. При цьому переходи між елементами, які відображаються здійснюються за допомогою анімації.

Android пропонує і інші реалізації View, такі, як елементи для вибору дати і часу, поля введення з автоматичним доповненням, галереї, вкладки і навіть карти (MapView).