**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

****

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №3**

**з курсу**

**«Архітектура мобільних операційних систем і пристроїв»**

*Студентки 4 курсу*

*групи ПП-42*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Матвіїв А.Ю.

*Викладач:*

Краснощок В.М.

**Київ – 2024**

***Тема:*** Робота із активностями, кольором та локалізацією в ОС Android.

***Мета:*** Отримати досвід роботи із процесами відстеження станів активності, використанням значень рядків і кольорів та зміною локалізації у додатку.

***Хід роботи:***

Добавимо даний код в activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

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

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="16dp"

android:text=""

android:textSize="40sp"

app:layout\_constraintBottom\_toTopOf="@+id/button"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent" />

<Button

android:id="@+id/button"

android:layout\_width="220dp"

android:layout\_height="120dp"

android:text="ПП-42"

android:textSize="24sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<ImageView

android:id="@+id/flagImageView"

android:layout\_width="308dp"

android:layout\_height="205dp"

android:layout\_below="@id/textView"

android:layout\_marginStart="48dp"

android:layout\_marginBottom="68dp"

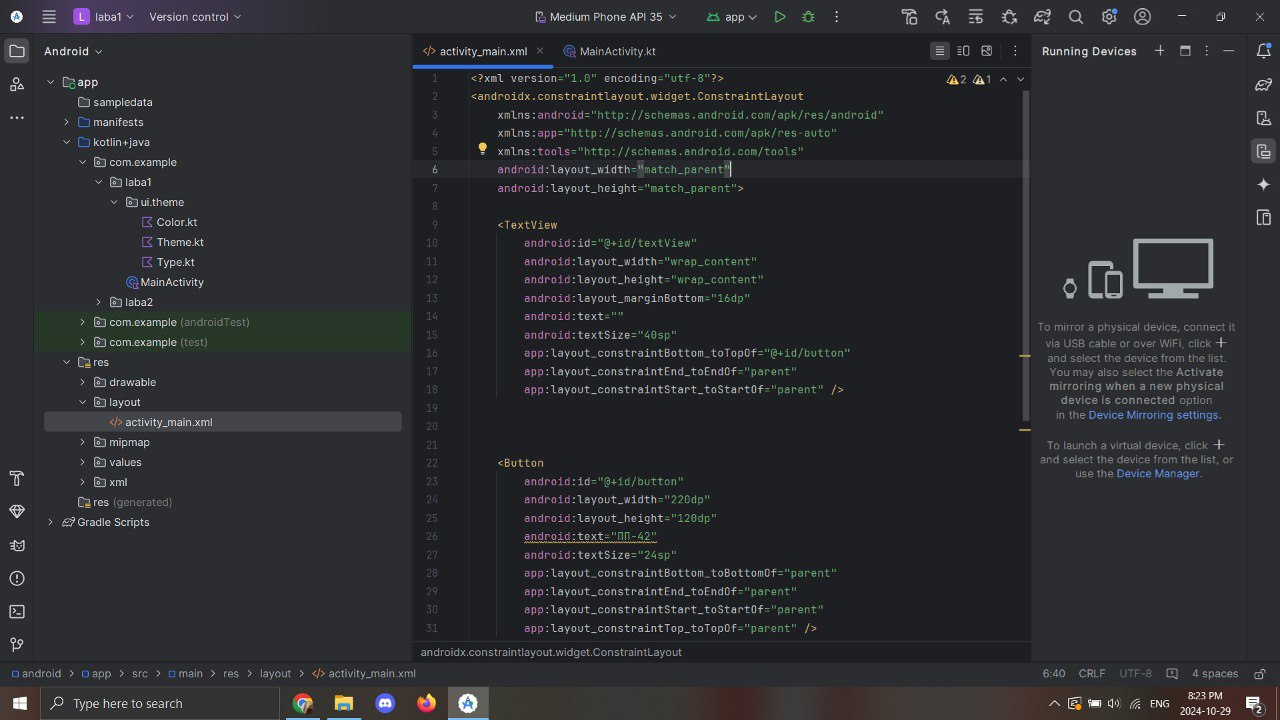
android:src="@drawable/flag\_default"

app:layout\_constraintBottom\_toBottomOf="@+id/textView"

app:layout\_constraintStart\_toStartOf="parent"

tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>



В Main.Activity добавимо код:  
package com.example.laba1

import android.annotation.SuppressLint

import android.graphics.Color

import android.os.Bundle

import android.util.Log

import android.widget.Button

import android.widget.ImageView

import android.widget.TextView

import androidx.activity.ComponentActivity

class MainActivity : ComponentActivity() {

private var isBackgroundBlue = false

@SuppressLint("SuspiciousIndentation")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val button: Button = findViewById(R.id.button)

val textView: TextView = findViewById(R.id.textView)

val flagImageView: ImageView = findViewById(R.id.flagImageView) // прапор

textView.text = ""

Log.d("ActivityLifecycle", "onCreate called")

val currentLocale = resources.configuration.locales[0]

when (currentLocale.language) {

"uk" -> {

flagImageView.setImageResource(R.drawable.flag\_ukraine)

Log.d("FlagUpdate", "Українська мова вибрана, прапор встановлено")

}

"pl" -> {

flagImageView.setImageResource(R.drawable.flag\_poland)

Log.d("FlagUpdate", "Польська мова вибрана, прапор встановлено")

}

else -> {

flagImageView.setImageResource(R.drawable.flag\_default)

Log.d("FlagUpdate", "Вибрано мову за замовчуванням, прапор встановлено")

}

}

button.setOnClickListener {

textView.text = "Anastasiia Matviiv"

textView.setBackgroundColor(Color.TRANSPARENT)

Log.d("ActivityState", "Text displayed: Anastasiia Matviiv, Background is blue: $isBackgroundBlue")

}

}

override fun onStart() {

super.onStart()

Log.d("ActivityLifecycle", "onStart called")

}

override fun onPause() {

super.onPause()

Log.d("ActivityLifecycle", "onPause called")

}

override fun onStop() {

super.onStop()

Log.d("ActivityLifecycle", "onStop called")

}

override fun onRestart() {

super.onRestart()

Log.d("ActivityLifecycle", "onRestart called")

}

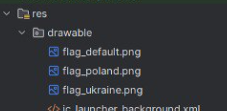
override fun onResume() {

super.onResume()

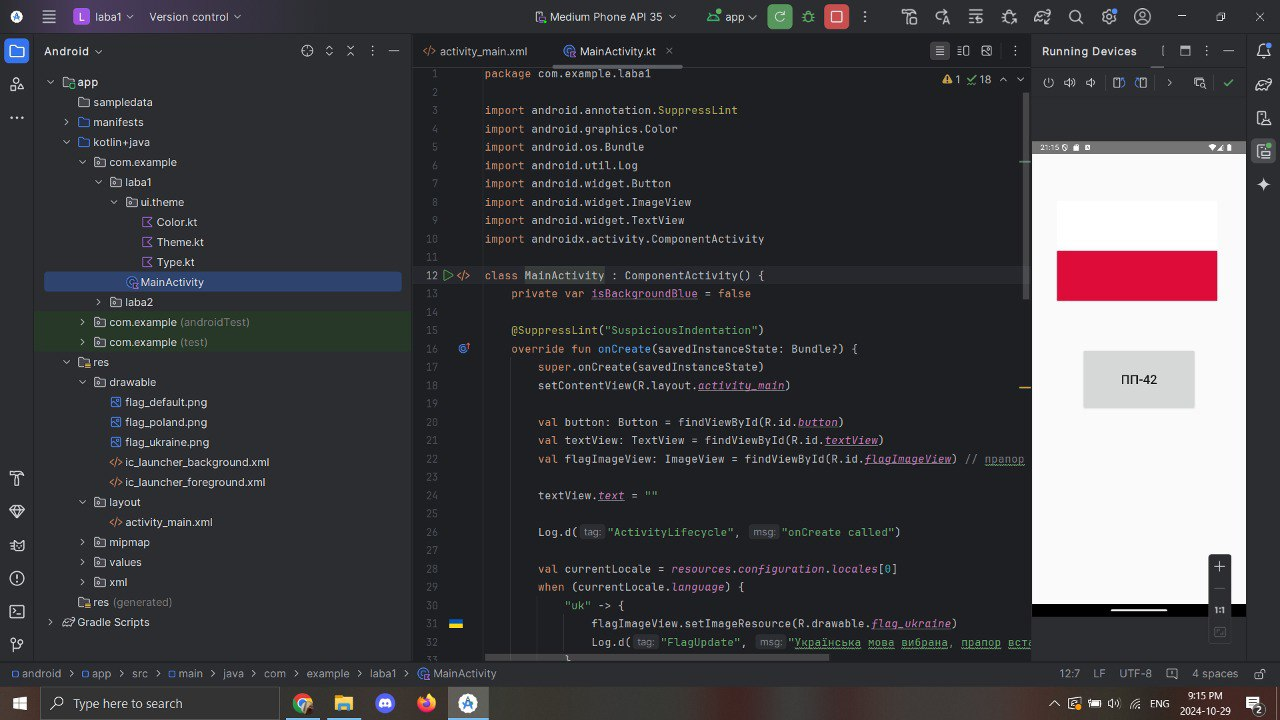
Log.d("ActivityLifecycle", "onResume called")

}

}

Також в папку drawable добавимо 3 прапори: український, польський і англійський (встановимо його по замовчуванню):  


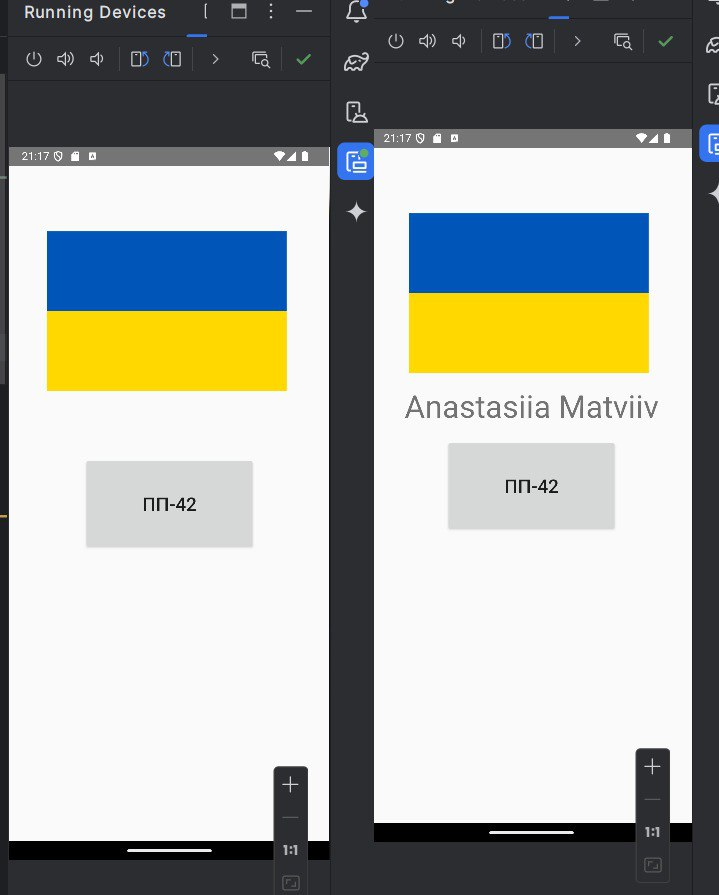
Приклади працювання при різних прапорах:



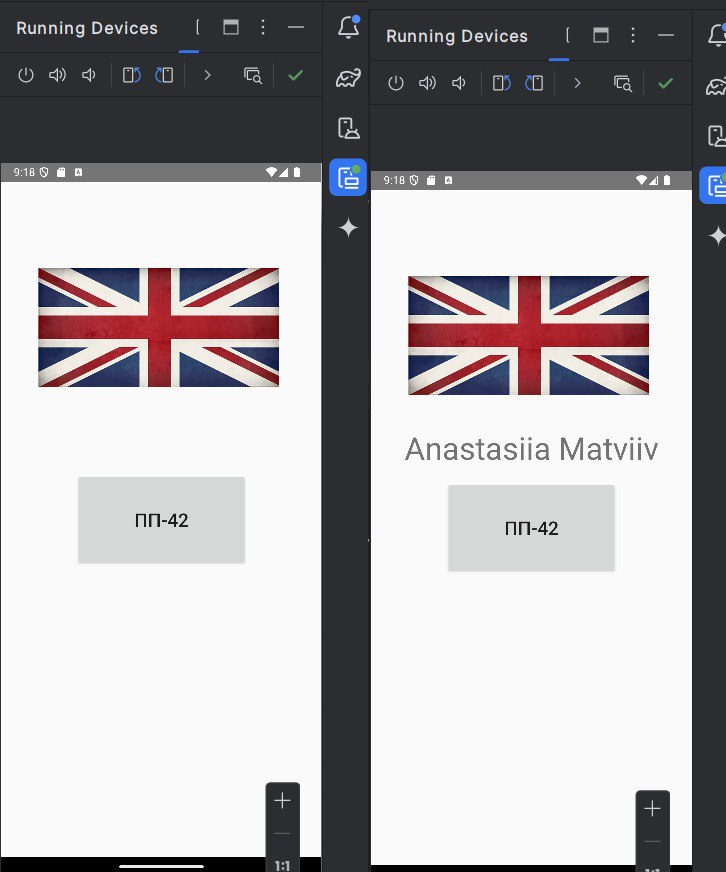
В самому емуляторі міняємо мову:



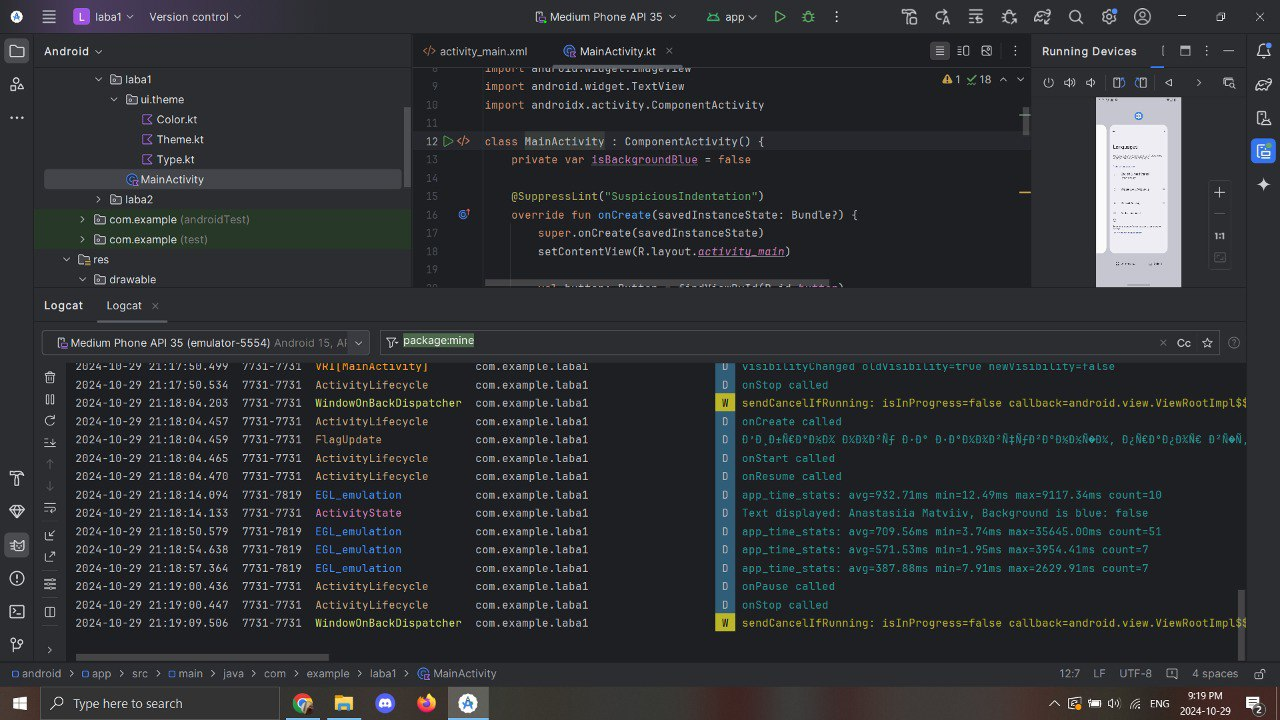
Українська мова:



Англійська мова:



Також можемо побачити що в logcat видно всі активності, в часності які прописали:



***Висновок:*** На даній лабораторній роботі я навчилася писати програми на Android Studio відповідно до обраної мови, отримала досвід роботи із процесами відстеження станів активності.

***Контрольні запитання:***

1. ***Стани активностей та їх відстеження.***

Активність в Android має кілька станів (створена, запущена, зупинена, знищена тощо), які визначають її видимість та доступність для взаємодії користувача. Відстеження цих станів дозволяє ефективно керувати ресурсами, зберігати дані та реагувати на зміни конфігурації пристрою.

Приклади методів:

* onCreate()
* onStart()
* onResume()
* onPause()
* onStop()
* onDestroy()

1. ***Особливості використання ресурсів у коді програми.***

Ресурси (зображення, рядки, макети) в Android зберігаються окремо від коду, що підвищує його читабельність та полегшує локалізацію. Використання R-класу дозволяє звертатися до ресурсів з коду.

1. ***Робота із анімацією в Android.***

Android підтримує різні типи анімацій (View Animation, Property Animation, Layout Transition), які дозволяють створювати плавні переходи та динамічні ефекти в інтерфейсі. Інструменти для створення анімацій: ObjectAnimator, ValueAnimator, AnimatorSet.

1. ***Робота із кольорами та колірні моделі в Android.***

Кольори в Android задаються у форматі ARGB, HEX або через ресурси. Колірні моделі включають sRGB та HSV. Використання кольорів дозволяє створювати привабливий та інтуїтивно зрозумілий інтерфейс.

1. ***Життєвий цикл активності.***

Життєвий цикл активності описує послідовність станів, через які проходить активність під час свого існування. Ключові методи життєвого циклу: onCreate(), onStart(), onResume(), onPause(), onStop(), onDestroy(). Розуміння життєвого циклу дозволяє правильно ініціалізувати компоненти, зберігати дані та звільняти ресурси.