

Отчет по лабораторной работе 5.2

Выполнили: Тютинина и Литвиненко

Группа: Ис231

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <WebView
        android:id="@+id/wwMain"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

MainActivity.kt:

```
package com.example.webviewapp

import android.os.Bundle
import android.webkit.WebView
import android.webkit.WebViewClient
import androidx.activity.OnBackPressedCallback
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var webView: WebView

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

        // Инициализация WebView
        webView = findViewById(R.id.wwMain)

        // Настройка WebView
        setupWebView()

        // Загрузка URL
        webView.loadUrl("https://github.com/")

        // Обработка кнопки "Назад" для навигации в WebView (современный способ)
        setupBackPressedHandler()
    }

    private fun setupWebView() {
        // Включение JavaScript (опционально)
    }
}
```

```

webView.settings.javaScriptEnabled = true

// Установка WebViewClient для обработки навигации внутри WebView
webView.webViewClient = object : WebViewClient() {
    // Этот метод гарантирует, что ссылки открываются внутри WebView
    override fun shouldOverrideUrlLoading(
        view: WebView?,
        url: String
    ): Boolean {
        view?.loadUrl(url)
        return true
    }
}

private fun setupBackPressedHandler() {
    // Современный способ обработки кнопки "Назад"
    onBackPressedDispatcher.addCallback(this, object :
OnBackPressedCallback(true) {
        override fun handleOnBackPressed() {
            if (webView.canGoBack()) {
                webView.goBack() // Возврат на предыдущую страницу в
WebView
            } else {
                isEnabled = false
                onBackPressedDispatcher.onBackPressed() // Выход из
приложения
            }
        }
    })
}
}

```

AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.webviewapp">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.WebViewApp">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

```
activity_main.xml x MainActivity.kt AndroidManifest.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical"
9     tools:context=".MainActivity">
10
11     <WebView
12         android:id="@+id/wwMain"
13         android:layout_width="match_parent"
14         android:layout_height="match_parent"/>
15
16 </LinearLayout>
```

```
<? activity_main.xml MainActivity.kt AndroidManifest.xml

1 package com.example.webviewapp
2
3 > import ...
8
2 Usages
class MainActivity : AppCompatActivity() {
10
6 Usages
11 private lateinit var webView: WebView
12
13 override fun onCreate(savedInstanceState: Bundle?) {
14     super.onCreate(savedInstanceState)
15     setContentView(R.layout.activity_main)
16
17     // Инициализация WebView
18     webView = findViewById(R.id.wvMain)
19
20     // Настройка WebView
21     setupWebView()
22
23     // Загрузка URL
24     webView.loadUrl(url = "https://github.com/")
25
26     // Обработка кнопки "Назад" для навигации в WebView (современный способ)
27     setupBackPressedHandler()
28 }
29
1 Usage
30 private fun setupWebView() {
31     // Включение JavaScript (опционально)
32     webView.settings.javaScriptEnabled = true
33
34     // Установка WebViewClient для обработки навигации внутри WebView
35     webView.webViewClient = object : WebViewClient() {
36         // Этот метод гарантирует, что ссылки открываются внутри WebView
37         override fun shouldOverrideUrlLoading(
38             view: WebView?,
39             url: String
40         ): Boolean {
41             view?.loadUrl(url)
42             return true
43         }
44     }
45 }
46
```

```
<? activity_main.xml MainActivity.kt AndroidManifest.xml

9 class MainActivity : AppCompatActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         // Обработка кнопки "Назад" для навигации в WebView (современный способ)
15         setupBackPressedHandler()
16     }
17
18     1 Usage
19     private fun setupWebView() {
20         // Включение JavaScript (опционально)
21         webView.settings.javaScriptEnabled = true
22
23         // Установка WebViewClient для обработки навигации внутри WebView
24         webView.webViewClient = object : WebViewClient() {
25             // Этот метод гарантирует, что ссылки открываются внутри WebView
26             override fun shouldOverrideUrlLoading(
27                 view: WebView?,
28                 url: String
29             ): Boolean {
30                 view?.loadUrl(url)
31                 return true
32             }
33         }
34     }
35
36     1 Usage
37     private fun setupBackPressedHandler() {
38         // Современный способ обработки кнопки "Назад"
39         onBackPressedDispatcher.addCallback(owner = this, onBackPressedCallback = object : OnBackPressedCallback(enabled = true) {
40             override fun handleOnBackPressed() {
41                 if (webView.canGoBack()) {
42                     webView.goBack() // Возврат на предыдущую страницу в WebView
43                 } else {
44                     isEnabled = false
45                     onBackPressedDispatcher.onBackPressed() // Выход из приложения
46                 }
47             }
48         })
49     }
50 }
51
```

```
activity_main.xml MainActivity.kt AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.webviewapp">
4
5     <uses-permission android:name="android.permission.INTERNET" />
6
7     <application
8         android:allowBackup="true"
9         android:icon="@mipmap/ic_launcher"
10        android:label="@string/app_name"
11        android:roundIcon="@mipmap/ic_launcher_round"
12        android:supportRtl="true"
13        android:theme="@style/Theme.WebViewApp">
14        <activity
15            android:name=".MainActivity"
16            android:exported="true">
17            <intent-filter>
18                <action android:name="android.intent.action.MAIN" />
19                <category android:name="android.intent.category.LAUNCHER" />
20            </intent-filter>
21        </activity>
22    </application>
23
24 </manifest>
```

Как работает сама программа:

