



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
**«Московский государственный технический университет имени Н.Э.  
Баумана (национальный исследовательский университет)»**  
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА №6**

**«Программирование виджетов рабочего экрана»**

**ДИСЦИПЛИНА: «Разработка мобильного ПО»**

Выполнил: студент гр. ИУК4-52Б

\_\_\_\_\_ (\_\_\_\_\_ Губин Е.В.\_\_\_\_\_)  
(Подпись) (Ф.И.О.)

Проверил:

\_\_\_\_\_ (\_\_\_\_\_ Прудяк П.Н.\_\_\_\_\_)  
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024 г.

**Цель:** формирование практических навыков создания пользовательских виджетов.

**Задачи:**

1. Научиться создавать пользовательские виджеты для рабочих экранов.
2. Научиться использовать конфигурационные окна для настройки виджетов.
3. Уметь понимать схемы взаимодействия виджета с другими элементами платформы Android.
4. Разработать эффективное приложение с учетом аппаратных ограничений мобильных устройств.
5. Научиться реализовывать логику работы приложения с учетом специфики платформы Android.

Формулировка задания:

**ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ**

Программа должна быть реализована на языке высокого уровня Kotlin.

9. Создать виджет – счетчик с текстовым полем и двумя кнопками. Текстовое поле отображает числа. При нажатии на первую кнопку значение текстового поля увеличиваются, а при нажатии на вторую кнопку, значения текстового поля уменьшаются.

**Листинг:**

CounterWidgetProvider:

```
package com.example.lw_6

import android.app.PendingIntent
import android.appwidget.AppWidgetManager
import android.appwidget.AppWidgetProvider
import android.content.ComponentName
import android.content.Context
import android.content.Intent
import android.widget.RemoteViews

class CounterWidgetProvider : AppWidgetProvider() {

    companion object {
        private const val ACTION_INCREMENT = "ACTION_INCREMENT"
```

```

private const val ACTION_DECREMENT = "ACTION_DECREMENT"
private const val PREFS_NAME = "CounterWidgetPrefs"
private const val COUNTER_KEY = "counter_value_"

private fun getCounterValue(context: Context, appWidgetId: Int): Int {
    val prefs = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE)
    return prefs.getInt(COUNTER_KEY + appWidgetId, 0)
}

private fun setCounterValue(context: Context, appWidgetId: Int, value: Int) {
    val prefs = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE)
    prefs.edit().putInt(COUNTER_KEY + appWidgetId, value).apply()
}

override fun onUpdate(context: Context, appWidgetManager: AppWidgetManager,
appWidgetIds: IntArray) {
    for (appWidgetId in appWidgetIds) {
        val views = RemoteViews(context.packageName, R.layout.widget_layout)

        val incrementIntent = Intent(context, CounterWidgetProvider::class.java).apply {
            action = ACTION_INCREMENT
            putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, appWidgetId)
        }
        val decrementIntent = Intent(context, CounterWidgetProvider::class.java).apply {
            action = ACTION_DECREMENT
            putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, appWidgetId)
        }

        val incrementPendingIntent = PendingIntent.getBroadcast(
            context,
            appWidgetId,
            incrementIntent,
            PendingIntent.FLAG_UPDATE_CURRENT or
PendingIntent.FLAG_IMMUTABLE
        )
        views.setOnClickPendingIntent(R.id.button_increment, incrementPendingIntent)

        val decrementPendingIntent = PendingIntent.getBroadcast(
            context,
            appWidgetId + 1,
            decrementIntent,
            PendingIntent.FLAG_UPDATE_CURRENT or
PendingIntent.FLAG_IMMUTABLE
        )
        views.setOnClickPendingIntent(R.id.button_decrement, decrementPendingIntent)

        val counterValue = getCounterValue(context, appWidgetId)

```

```

        views.setTextViewText(R.id.text_counter, counterValue.toString())

        appWidgetManager.updateAppWidget(appWidgetId, views)
    }
}

override fun onReceive(context: Context, intent: Intent) {
    super.onReceive(context, intent)
    val appWidgetId = intent.getIntExtra(AppWidgetManager.EXTRA_APPWIDGET_ID,
-1)
    if (appWidgetId != -1 && (intent.action == ACTION_INCREMENT || intent.action ==
ACTION_DECREMENT)) {
        val counterValue = getCounterValue(context, appWidgetId)
        val newValue = when (intent.action) {
            ACTION_INCREMENT -> counterValue + 1
            ACTION_DECREMENT -> counterValue - 1
            else -> counterValue
        }
        setCounterValue(context, appWidgetId, newValue)

        val appWidgetManager = AppWidgetManager.getInstance(context)
        val views = RemoteViews(context.packageName, R.layout.widget_layout)
        views.setTextViewText(R.id.text_counter, newValue.toString())
        appWidgetManager.updateAppWidget(appWidgetId, views)
    }
}
}

```

widget\_layout.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:background="@android:color/white"
    android:padding="8dp">

    <Button
        android:id="@+id/button_decrement"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-" />

    <TextView
        android:id="@+id/text_counter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="16dp"

```

```

        android:text="0"
        android:textSize="24sp" />

<Button
    android:id="@+id/button_increment"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="+" />
</LinearLayout>

```

counter\_widget.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:initialLayout="@layout/widget_layout"
    android:minWidth="200dp"
    android:minHeight="100dp"
    android:updatePeriodMillis="0" />

```

AndroidManifest.xml:

```

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LW_6"
        tools:targetApi="31">

        <receiver
            android:name=".CounterWidgetProvider"
            android:label="Counter Widget"
            android:exported="false">

            <intent-filter>
                <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
            </intent-filter>

            <meta-data
                android:name="android.appwidget.provider"
                android:resource="@xml/counter_widget" />

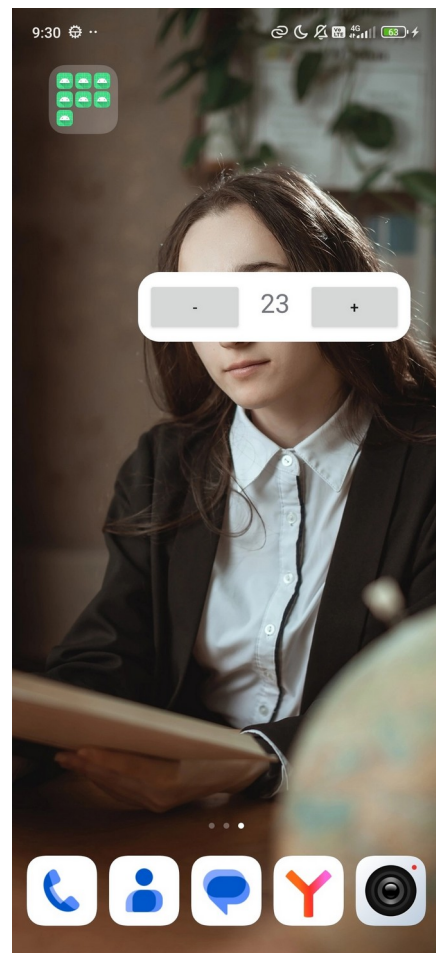
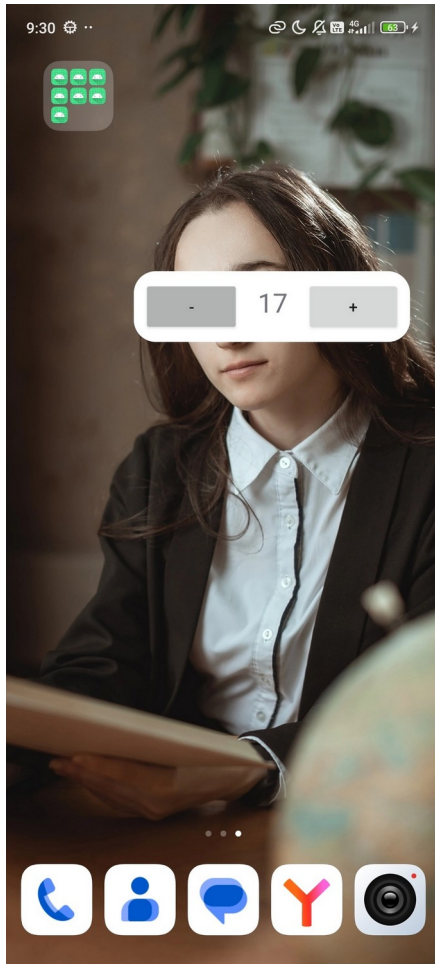
```

```
</receiver>

</application>

</manifest>
```

### Результаты выполнения работы:



**Вывод:** в ходе оабораторной работы был реализован счётчик в виде виджета рабочего экрана.