



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 №7

ДИСЦИПЛИНА: «Разработка мобильного программного обеспечения»

Выполнил: студент гр. ИУК4-62Б \_\_\_\_\_ (Подпись) (Калашников А.С.)  
(Ф.И.О.)

Проверил: \_\_\_\_\_ (Подпись) (Шаматрин А.Г.)  
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2023

**Целью** выполнения лабораторной работы является формирование практических навыков создания пользовательских виджетов.

**Основными задачами** выполнения лабораторной работы являются:

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.

### **Вариант №7**

#### **Задание:**

Создать виджет отображающий текущую дату: День недели и месяц должны отображаться на русском языке. По нажатию на виджет должно открываться окно конфигурации для настройки цвета фона и шрифта. Цвет выбирается по клику на ImageButton.

#### **Листинг реализованной программы:**

##### **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">

    <queries>
        <package android:name="com.example.carbasedata" />
    </queries>

    <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:supportsRtl="true"
        android:theme="@style/Theme.LR_6"
        tools:targetApi="31">
        <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>
```

##### **MyWidgetConfigureActivity.kt**

```

package com.example.myapplication

import android.app.Activity
import android.appwidget.AppWidgetManager
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.view.View
import android.widget.EditText
import android.widget.ImageButton
import com.example.myapplication.databinding.MyWidgetConfigureBinding

/**
 * The configuration screen for the [MyWidget] AppWidget.
 */
class MyWidgetConfigureActivity : Activity() {

    private var appWidgetId = AppWidgetManager.INVALID_APPWIDGET_ID
    var text_color: String = "FFFFFF"
    var color: String = "000000"
    private lateinit var appWidgetText: EditText
    private var onClickListener = View.OnClickListener {
        val context = this@MyWidgetConfigureActivity

        // When the button is clicked, store the string locally
        val widgetText = appWidgetText.text.toString()
        saveTitlePref(context, appWidgetId, widgetText)

        // It is the responsibility of the configuration activity to update the
        app widget

        // Make sure we pass back the original appWidgetId
        val resultValue = Intent()
        resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, appWidgetId)
        resultValue.putExtra("color", color)
        resultValue.putExtra("color_text", text_color)
        setResult(RESULT_OK, resultValue)
        finish()
    }
    private lateinit var binding: MyWidgetConfigureBinding

    public override fun onCreate( savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Set the result to CANCELED. This will cause the widget host to cancel
        // out of the widget placement if the user presses the back button.
        //setResult(RESULT_CANCELED)

        binding = MyWidgetConfigureBinding.inflate(layoutInflater)
        setContentView(binding.root)

        appWidgetText = binding.appwidgetText as EditText
        binding.addButton.setOnClickListener(onClickListener)

        // Find the widget id from the intent.
        val appWidgetId =
            intent?.extras?.getInt(AppWidgetManager.EXTRA_APPWIDGET_ID,
                AppWidgetManager.INVALID_APPWIDGET_ID) ?: AppWidgetManager.INVALID_APPWIDGET_ID

        // If this activity was started with an intent without an app widget ID,
        finish with an error.
        if (appWidgetId == AppWidgetManager.INVALID_APPWIDGET_ID) {
            finish()
            return
        }
    }
}

```

```

    }
    val imageButton = findViewById<ImageButton>(R.id.imageButton)
    val imageButton2 = findViewById<ImageButton>(R.id.imageButton2)
    val imageButton5 = findViewById<ImageButton>(R.id.imageButton5)
    val imageButton6 = findViewById<ImageButton>(R.id.imageButton6)
    imageButton.setOnClickListener {
        color = "FFBB86FC"
    }
    imageButton2.setOnClickListener {
        color = "000000"
    }
    imageButton5.setOnClickListener {
        text_color = "FFFFFF"
    }
    imageButton6.setOnClickListener {
        text_color = "000000"
    }
    val resultValue = Intent()
    resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, appWidgetId)
    resultValue.putExtra("color", color)
    resultValue.putExtra("text_color", text_color)
    setResult(RESULT_OK, resultValue)

    //      val appWidgetManager = AppWidgetManager.getInstance(applicationContext)
    //      appWidgetManager.updateAppWidget(appWidgetId,
MyWidget.buildRemoteViews(applicationContext, appWidgetId))
    //      appWidgetManager.notifyAppWidgetViewDataChanged(appWidgetId,
R.id.layout_widget)
    //
    //      appWidgetText.setText(loadTitlePref(this@MyWidgetConfigureActivity,
appWidgetId))
    }

}

private const val PREFS_NAME = "com.example.myapplication.MyWidget"
private const val PREF_PREFIX_KEY = "appwidget_"

// Write the prefix to the SharedPreferences object for this widget
internal fun saveTitlePref(context: Context, appWidgetId: Int, text: String) {
    val prefs = context.getSharedPreferences(PREFS_NAME, 0).edit()
    prefs.putString(PREF_PREFIX_KEY + appWidgetId, text)
    prefs.apply()
}

// Read the prefix from the SharedPreferences object for this widget.
// If there is no preference saved, get the default from a resource
internal fun loadTitlePref(context: Context, appWidgetId: Int): String {
    val prefs = context.getSharedPreferences(PREFS_NAME, 0)
    val titleValue = prefs.getString(PREF_PREFIX_KEY + appWidgetId, null)
    return titleValue ?: context.getString(R.string.appwidget_text)
}

internal fun deleteTitlePref(context: Context, appWidgetId: Int) {
    val prefs = context.getSharedPreferences(PREFS_NAME, 0).edit()
    prefs.remove(PREF_PREFIX_KEY + appWidgetId)
    prefs.apply()
}

```

## MyWidget.kt

```
package com.example.myapplication
```

```
import android.app.PendingIntent
import android.appwidget.AppWidgetManager
```

```

import android.appwidget.AppWidgetProvider
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.util.Log
import android.widget.RemoteViews
import android.widget.Toast
import java.text.SimpleDateFormat
import java.util.*

/**
 * Implementation of App Widget functionality.
 * App Widget Configuration implemented in [MyWidgetConfigureActivity]
 */
class MyWidget : AppWidgetProvider() {
    private val myReceiver = object: BroadcastReceiver() {
        override fun onReceive(context: Context?, intent: Intent?) {
            val value = intent?.getIntExtra("color", 0) ?: 0
            // используйте значение для обновления вашего виджета
        }
    }

    override fun onUpdate(
        context: Context,
        appWidgetManager: AppWidgetManager,
        appWidgetIds: IntArray
    ) {
        // There may be multiple widgets active, so update all of them
        for (appWidgetId in appWidgetIds) {
            this.updateAppWidget(context, appWidgetManager, appWidgetId)
        }

        super.onUpdate(context, appWidgetManager, appWidgetIds)
    }

    override fun onDeleted(context: Context, appWidgetIds: IntArray) {
        // When the user deletes the widget, delete the preference associated with
it.
        for (appWidgetId in appWidgetIds) {
            deleteTitlePref(context, appWidgetId)
        }
    }

    override fun onEnabled(context: Context) {
        // Enter relevant functionality for when the first widget is created
    }

    override fun onDisabled(context: Context) {
        // Enter relevant functionality for when the last widget is disabled
    }

    override fun onReceive(context: Context?, intent: Intent?) {
        super.onReceive(context, intent)
    }

    companion object {
        fun buildRemoteViews(context: Context, appWidgetId: Int): RemoteViews {
            val widgetText = loadTitlePref(context, appWidgetId)
            // Construct the RemoteViews object
            val views = RemoteViews(context.packageName, R.layout.my_widget)
            val configIntent = Intent(context,
MyWidgetConfigureActivity::class.java)
            configIntent.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID,
appWidgetId)

            val configPendingIntent = PendingIntent.getActivity(context,

```

```

appWidgetId, configIntent, 0)
    views.setOnClickPendingIntent(R.id.layout_widget, configPendingIntent)
    configIntent.action =
"android.appwidget.action.APPWIDGET_CONFIGURE$appWidgetId"

    val dateFormat = SimpleDateFormat("d MMM yyyy, EEE")
    val date = dateFormat.format(Date())
    views.setTextViewText(R.id.appwidget_text, date)
    return views
}

private fun updateAppWidget(
    context: Context,
    appWidgetManager: AppWidgetManager,
    appWidgetId: Int
) {
    val views = buildRemoteViews(context, appWidgetId)

    // Instruct the widget manager to update the widget
    appWidgetManager.updateAppWidget(appWidgetId, views)
}
}

```

### **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"
tools:context=".MainActivity">

<EditText
    android:id="@+id/editText"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_marginTop="141dp"
    android:text="Stamp"
    app:layout_constraintEnd_toEndOf="@+id/editText2"
    app:layout_constraintStart_toStartOf="@+id/editText2"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/editText2"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_marginTop="90dp"
    android:layout_marginBottom="95dp"
    android:text="Model"
    app:layout_constraintBottom_toBottomOf="@+id/editText3"
    app:layout_constraintEnd_toEndOf="@+id/editText3"
    app:layout_constraintStart_toStartOf="@+id/editText3"
    app:layout_constraintTop_toTopOf="@+id/editText" />

<EditText
    android:id="@+id/editText3"
    android:layout_width="200dp"
    android:layout_height="50dp"
    android:layout_marginStart="88dp"
    android:layout_marginBottom="64dp"
    android:text="Year"
    app:layout_constraintBottom_toTopOf="@+id/button"

```

```

        app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="19dp"
    android:text="Далее"
    app:layout_constraintBaseline_toBaselineOf="@+id/button2"
    app:layout_constraintEnd_toStartOf="@+id/button2"
    app:layout_constraintHorizontal_chainStyle="spread_inside"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="4dp"
    android:layout_marginBottom="40dp"
    android:text="В начало"
    app:layout_constraintBottom_toTopOf="@+id/button5"
    app:layout_constraintEnd_toStartOf="@+id/button3"
    app:layout_constraintStart_toEndOf="@+id/button" />

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="16dp"
    android:text="Добавить"
    app:layout_constraintBaseline_toBaselineOf="@+id/button2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/button2" />

<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="19dp"
    android:text="Удалить"
    app:layout_constraintBaseline_toBaselineOf="@+id/button5"
    app:layout_constraintEnd_toStartOf="@+id/button5"
    app:layout_constraintHorizontal_chainStyle="spread_inside"
    app:layout_constraintStart_toStartOf="parent" />

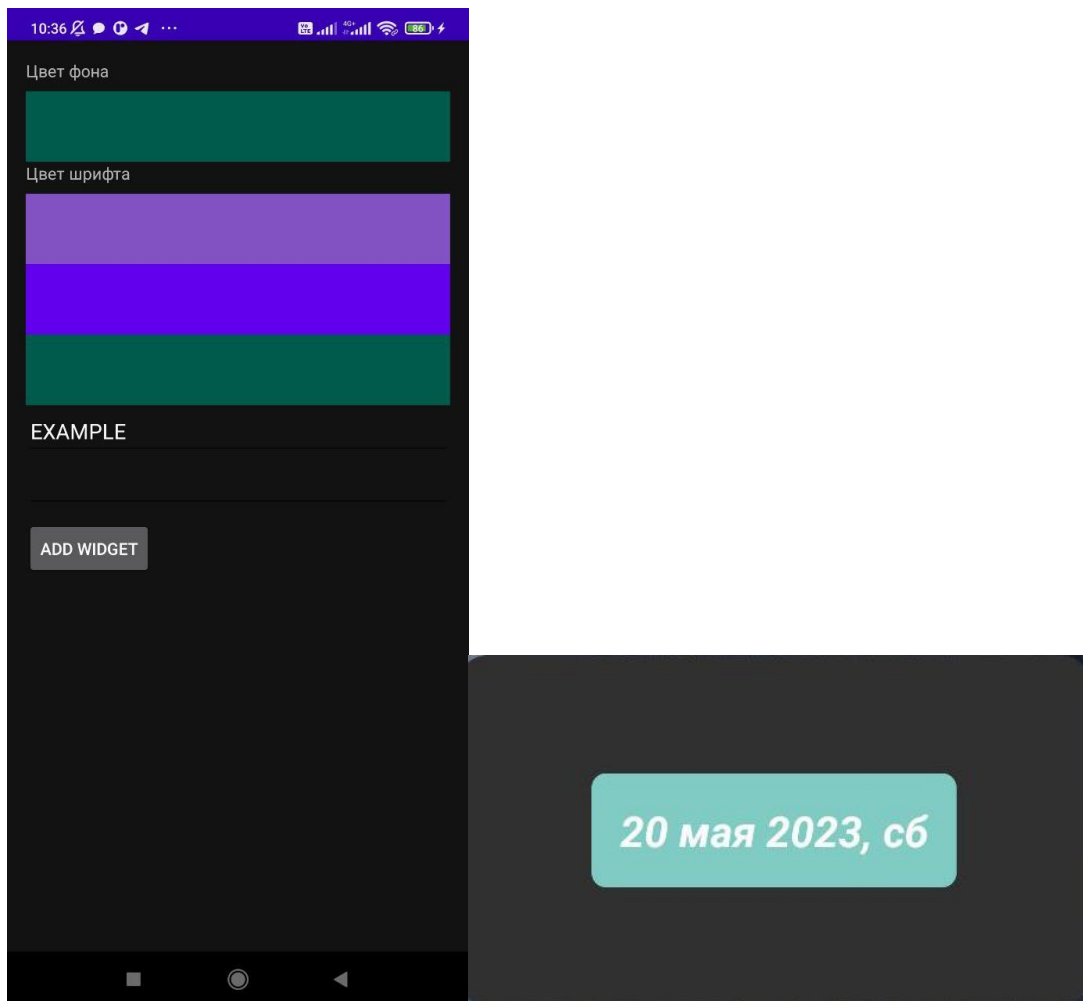
<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="4dp"
    android:layout_marginBottom="155dp"
    android:text="Обновить"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/button6"
    app:layout_constraintStart_toEndOf="@+id/button4" />

<Button
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="16dp"
    android:text="Очистить"
    app:layout_constraintBaseline_toBaselineOf="@+id/button5"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/button5" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

### Результат выполнения работы:



**Рис. 1.** Работа программы

**Вывод:** в ходе выполнения данной лабораторной работы были сформированы навыки создания пользовательских виджетов.