|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Калашников А.С.)  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Шаматрин А.Г.)  (Подпись) (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>

**MyWidgetConigureActivity.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(icicle: Bundle?) {  
 super.onCreate(icicle)  
  
 // 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()  
}

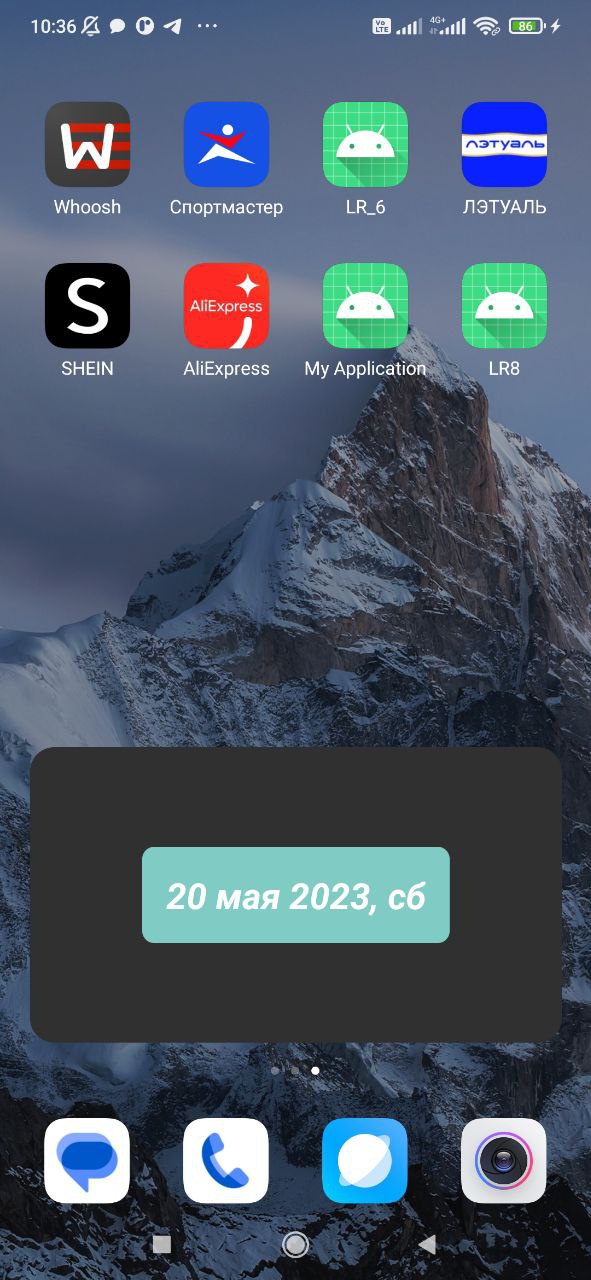
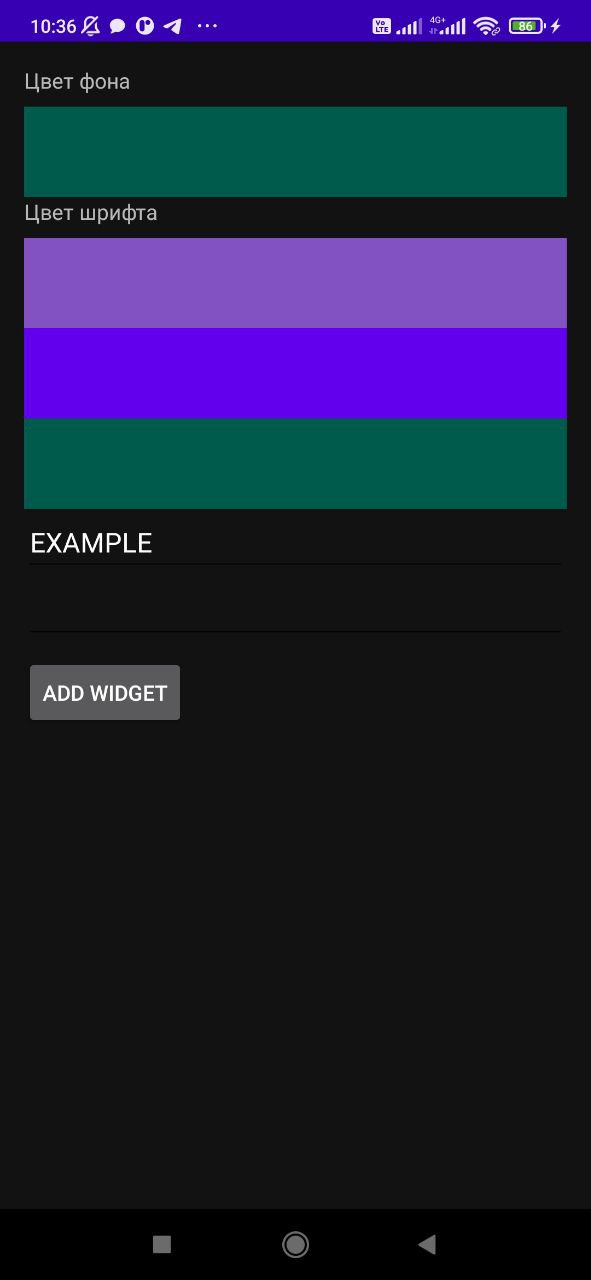
**MyWidget.kt**

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.