

## **Разработка Android-приложения с поддержкой мультиязычного интерфейса**

*Андриенко Иван Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема*  
*Студент*

*Пащенко Екатерина Александровна*

*Приамурский государственный университет имени Шолом-Алейхема*  
*Студент*

### **Аннотация**

В статье рассматривается процесс создания Android-приложения, поддерживающего мультиязычный интерфейс с возможностью динамического переключения языка пользователем. Описаны ключевые этапы реализации локализации, включая использование ресурсов strings.xml для различных языков, настройку пользовательского интерфейса, а также механизм смены языка в рантайме с применением Java. Представленный подход позволяет разработать гибкое и удобное приложение, адаптированное под международную аудиторию. Реализация выполнена в среде Android Studio с использованием языка программирования Java и стандартных средств Android SDK.

**Ключевые слова:** Android, Java, мультиязычность, локализация, интерфейс, strings.xml, ресурсы, мобильная разработка

### **Development of an Android application with multilingual interface support**

*Andrienko Ivan Sergeevich*

*Sholom-Aleichem Priamursky State University*  
*Student*

*Paschenko Ekaterina Alexandrovna*

*Sholom-Aleichem Priamursky State University*  
*Student*

### **Abstract**

The article discusses the process of creating an Android application that supports a multilingual interface with the ability to dynamically switch the language by the user. The key stages of localization implementation, including resource usage, are described. strings.xml for different languages, customizing the user interface, as well as a mechanism for changing the language in runtime using Java. The presented approach makes it possible to develop a flexible and user-friendly application adapted to an international audience. The implementation is performed in the

Android Studio environment using the Java programming language and standard Android SDK tools.

**Keywords:** Android, Java, multilingualism, localization, interface, strings.xml , resources, mobile development

## **1 Введение**

### **1.1 Актуальность**

Современные мобильные приложения всё чаще ориентированы на международные рынки, что делает поддержку нескольких языков неотъемлемой частью пользовательского опыта. Возможность адаптировать интерфейс под язык системы или индивидуальные предпочтения пользователя значительно повышает удобство использования и расширяет аудиторию приложения. Тем не менее, реализация мультиязычности требует понимания архитектуры Android и правильной работы с ресурсами, что делает данную тему актуальной для разработчиков и исследователей в области мобильных технологий.

### **1.2 Обзор исследований**

Проблемы разработки визуальных компонентов для Android-приложений активно исследуются в научной литературе. В работе Д.В.Тюкавина и И.В. Смагиной проведен анализ программного обеспечения для проектирования дизайна мобильных приложений, где особое внимание уделено инструментам создания анимированных интерфейсов [1]. Н.С.Андриенко в своем исследовании рассматривает ключевые аспекты разработки визуального дизайна Android-приложений, включая использование кастомных View-компонентов для реализации динамических эффектов [2]. Методические аспекты проектирования интерфейсов в стиле Material Design изучены А.А. Ивановым и соавторами. В их работе представлены рекомендации по созданию анимированных элементов, обеспечивающих плавность взаимодействия с пользователем [3]. В.С.Богданова акцентирует внимание на процессе проектирования дизайна Android-приложений, включая разработку фоновых компонентов с учетом современных тенденций UX/UI [4]. Философские основы дизайна интерфейсов для мобильных платформ рассмотрены Е.А. Шумиловой и А.В. Свищёвым. В их исследовании подчеркивается важность анимации как средства повышения пользовательской вовлеченности и визуальной гармонии приложения [5].

### **1.3 Цель исследования**

Цель исследования – разработка Android-приложения на языке Java с поддержкой мультиязычного интерфейса. В рамках реализации рассматриваются методы локализации ресурсов, переключения языка интерфейса в рантайме, а также сохранения выбранных настроек. Проект направлен на демонстрацию практического подхода к созданию универсальных и адаптируемых пользовательских интерфейсов.

## **2 Материалы и методы**

Для достижения поставленной цели использовались среда разработки Android Studio версии 2022.3.1, язык программирования Java, а также устройство с операционной системой Android версии 8.0 и выше для тестирования. В качестве основного источника информации использовалась документация Android Developers.

## **3 Результаты и обсуждения**

Поддержка мультязычного интерфейса в Android-приложениях базируется на механизме ресурсных файлов. Все текстовые элементы выносятся в отдельный файл `strings.xml`, размещаемый в папке `res/values`. Для добавления новых языков создаются дополнительные папки вида `values-ru`, `values-es` и другие, в зависимости от кода языка, где размещаются альтернативные версии `strings.xml` с переведёнными строками. Android автоматически подгружает ресурсы соответствующего языка в зависимости от системных настроек устройства. Таким образом, при запуске приложения пользователь видит интерфейс на том языке, который установлен в системе.

Для реализации ручного переключения языка в реальном времени требуется переопределение языка внутри активности. На этапе инициализации приложения создаётся метод `setLocale`, который получает строку языка (например, `ru` или `en`) и устанавливает соответствующий язык с помощью класса `Locale` и объекта `Configuration`. После установки вызывается метод `recreate()` для перезагрузки текущей активности и применения новых языковых ресурсов (рис. 1).

```
13 </> public class MainActivity extends AppCompatActivity {  
14  
15     private TextView textHello; 1 usage  
16     private Button buttonChange; 2 usages  
17  
18     @Override  
19     protected void onCreate(Bundle savedInstanceState) {  
20         super.onCreate(savedInstanceState);  
21         setLocale(Locale.getDefault().getLanguage());  
22         setContentView(R.layout.activity_main);  
23  
24         textHello = findViewById(R.id.textHello);  
25         buttonChange = findViewById(R.id.buttonChangeLanguage);  
26  
27         buttonChange.setOnClickListener(new View.OnClickListener() {  
28             @Override  
29             public void onClick(View v) {  
30                 switchLanguage();  
31             }  
32         });  
33     }  
34  
35     private void switchLanguage() { 1 usage  
36         String currentLang = Locale.getDefault().getLanguage();  
37         String newLang = currentLang.equals("ru") ? "en" : "ru";  
38         setLocale(newLang);  
39  
40         recreate(); // Перезапускаем активность  
41     }  
42  
43     private void setLocale(String lang) { 2 usages  
44         Locale locale = new Locale(lang);  
45         Locale.setDefault(locale);  
46         Configuration config = new Configuration();  
47         config.setLocale(locale);  
48         getResources().updateConfiguration(config, getResources().getDisplayMetrics());  
49     }  
50 }  
51
```

Рис. 1. Код для переключения языка

Создание многоязычного интерфейса в мобильном приложении на платформе Android начинается с организации текстовых ресурсов, отвечающих за отображение надписей на экране. Все текстовые элементы, которые должны выводиться в программе, выносятся в специальный файл, содержащий набор строк. По умолчанию такой файл размещается в основной папке ресурсов и используется для базового языка интерфейса – например, английского или русского.

Чтобы приложение отображало надписи на разных языках в зависимости от настроек устройства, необходимо создать отдельные папки с переводами. Каждая такая папка соответствует определённому языку и содержит свой вариант файла с текстовыми строками. Для создания папки с русскоязычным интерфейсом в среде разработки Android Studio используется встроенный мастер. Сначала открывается структура проекта, где отображаются все ресурсы приложения. После этого выполняется добавление новой папки с помощью команды создания ресурса. В открывшемся окне выбирается тип

создаваемой папки – текстовые значения, и указывается нужный язык – русский. В результате автоматически создаётся папка, предназначенная для русских текстов (рис. 2).

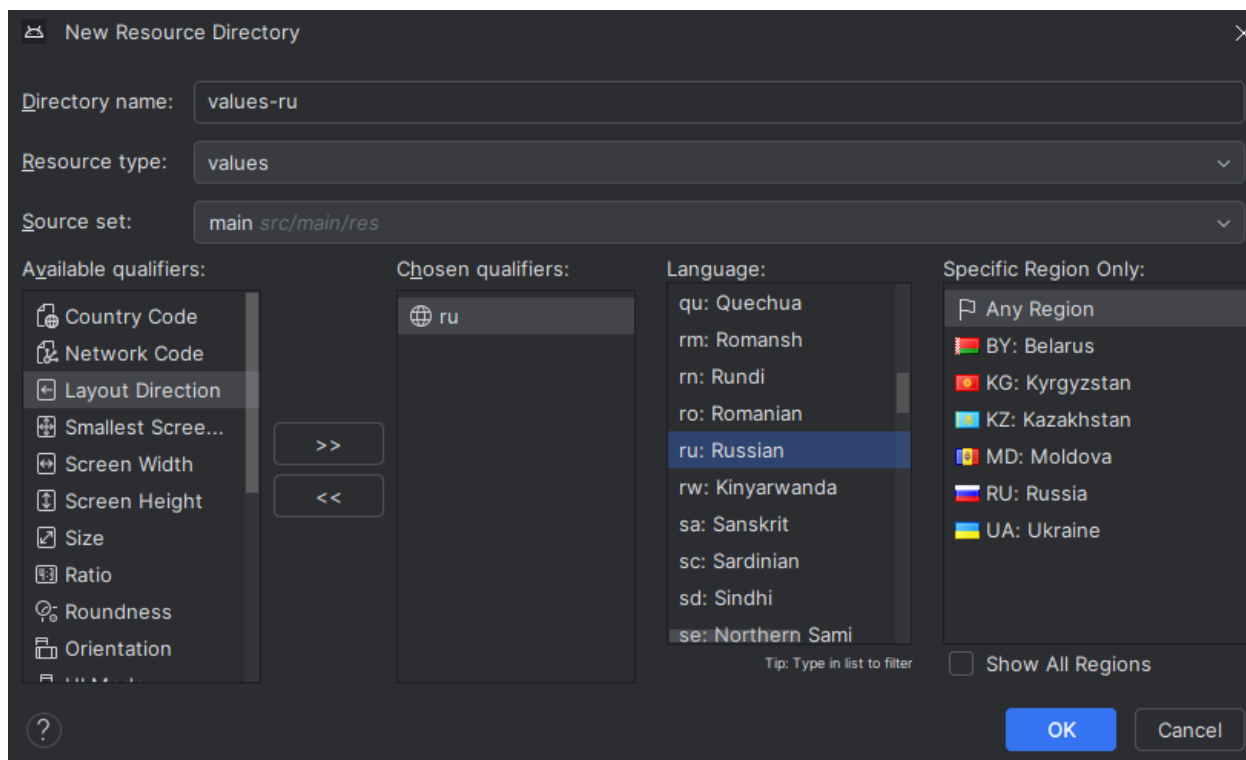


Рис. 2. Создание папки с русскоязычным интерфейсом

Внутри созданной папки необходимо добавить файл с тем же названием, что и основной – обычно это `strings.xml`. В этот файл помещаются переводы всех строк, используемых в приложении. Важно, чтобы названия строк совпадали с теми, что заданы в оригинальном файле, иначе подстановка текста не произойдёт. При запуске приложения система автоматически определяет, какой язык установлен на устройстве, и подгружает соответствующий набор текстов. Это позволяет без дополнительных усилий отображать интерфейс на нужном языке (рис. 3).

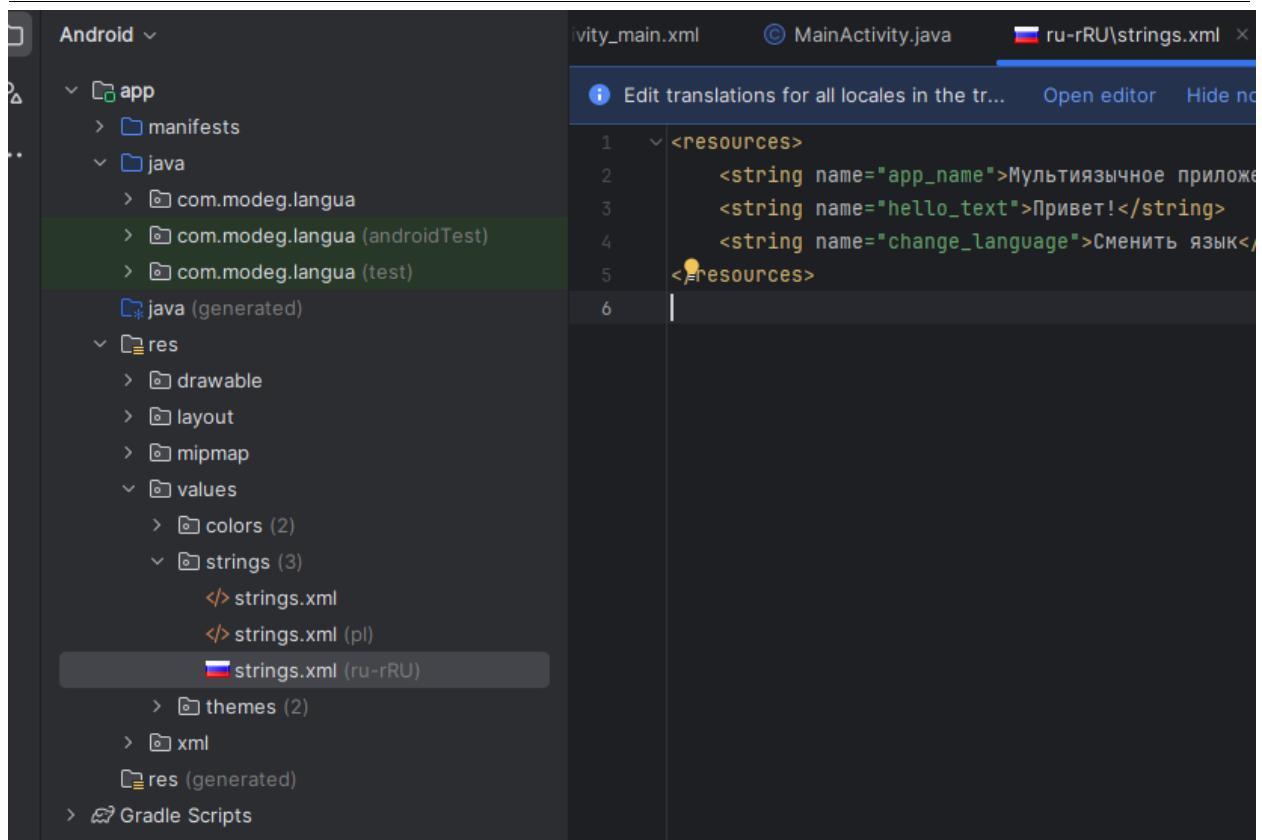


Рис. 3. Создание русскоязычного strings.xml

В ходе тестирования приложение корректно определяло язык системы и запускалось сразу с нужным интерфейсом. При включённом русском языке все надписи отображались на русском, при английском – на английском. Смена языка вручную происходила мгновенно. После нажатия на кнопку все тексты обновлялись без задержек и без необходимости перезапуска приложения. Интерфейс корректно переключался как с русского на английский, так и обратно. Повторный запуск сохранял выбранный язык, и интерфейс сразу отображался на нём (рис. 4).

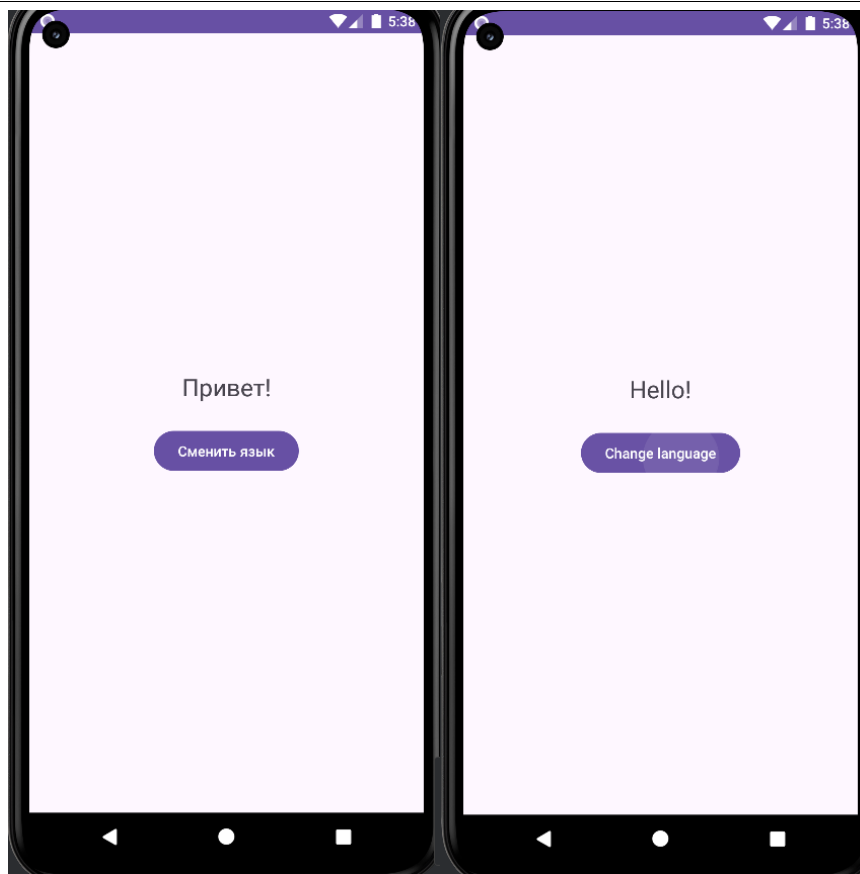


Рис. 4. Тестирование смены языка

#### 4 Выводы

В результате работы было разработано Android-приложение с поддержкой многоязычного интерфейса. Реализован механизм автоматического определения языка системы, а также ручного переключения языка во время работы программы. Все изменения отображаются в реальном времени без перезапуска, что повышает удобство для пользователя. Предложенный подход позволяет адаптировать мобильное приложение под различные языковые аудитории и может быть использован при создании универсальных пользовательских интерфейсов.

#### Библиографический список

1. Тюкавин Д.В., Смагина И.В. Анализ программного обеспечения для разработки дизайна и программирования мобильного приложения // Экономическая среда. 2019. № 2 (28). С. 5-9.
2. Андриенко Н.С. Разработка визуального дизайна Android-приложения // В сборнике: Молодые исследователи - регионам. Материалы Международной научной конференции. В 3-х томах / Отв. ред. А.А. Сеницын. 2018. С. 36-37.
3. Иванов А.А., Колташёва Д.Д., Сардак Л.В. Методические рекомендации по разработке дизайна в стиле Material Design для мобильного приложения под ОС Android // Актуальные вопросы преподавания математики,

- информатики и информационных технологий. 2019. № 4. С. 238-241.
4. Богданова В.С. Проектирование дизайна Android приложения // В сборнике: Приоритетные направления развития науки в современном мире. Сборник научных статей по материалам XVI Международной научно-практической конференции. Уфа, 2025. С. 159-162.
  5. Шумилова Е.А., Свищёв А.В. Философия и принципы дизайна интерфейсов платформ Android и iOS // Моя профессиональная карьера. 2023. Т. 2. № 54. С. 270-277.