

## Разработка мобильного приложения для генерации QR кодов

*Андрюченко Иван Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье представлено подробное пошаговое руководство по разработке мобильного приложения для генерации QR-кодов с использованием Android Studio и языка программирования Java. Подробно описан процесс создания пользовательского интерфейса, интеграции библиотеки ZXing для работы с QR-кодами, а также добавление функционала, которое позволяет пользователям вводить свой собственный текст перед генерацией кода. Результатом работы станет простое и функциональное приложение, готовое к использованию для создания QR-кодов на устройствах под управлением Android.

**Ключевые слова:** Android, QR-код, мобильное приложение, Android Studio, Java, пользовательский интерфейс, ZXing.

## Development of a mobile application for generating QR codes

*Andrienko Ivan Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

This article provides a detailed step-by-step guide to developing a mobile application for generating QR codes using Android Studio and the Java programming language. The process of creating a user interface, integrating the ZXing library for working with QR codes, as well as adding functionality that allows users to enter their own text before generating the code is described in detail. The result of the work will be a simple and functional application ready to use for creating QR codes on Android devices.

**Keywords:** Android, QR code, Mobile application, Android Studio, Java, User interface, ZXing.

## 1 Введение

### 1.1 Актуальность

В наше время QR-коды становятся все более важным инструментом для передачи информации и удобным способом взаимодействия с данными. Разработка мобильного приложения для генерации QR-кодов в Android Studio предоставляет уникальную возможность освоить основы создания приложений под платформу Android и расширить навыки программирования

на языке Java. С учетом широкого применения QR-кодов в различных областях, от бизнеса до личного использования, разработка такого приложения может быть полезна как для новичков, так и для опытных разработчиков, стремящихся овладеть этой технологией. Познавание процесса генерации QR-кодов и взаимодействия с пользовательским вводом открывает двери к созданию практичных приложений с широким спектром применения.

### **1.2 Обзор исследований**

В своей работе А.С. Кожевникова рассмотрела разработку клиент-серверного приложения для создания и распознавания QR кода с электронной цифровой подписью при использовании современных популярных фреймворков (Maven, Spring) для отображения информации о сотрудниках компании, которая подгружается посредством подключения к базе данных PostgreSQL [1]. З.Э. Халилова рассмотрела основные причины использования QR-кода, который позволяет представить большое количество информации различного рода в виде последовательности символов, обеспечивает удобство использования. Кроме того, QR-коды позволяют считывать информацию с помощью камеры устройства. Также на базе разработки Android-приложений показаны приемы генерации новых и сканирования существующих QR-кодов с использованием библиотеки с открытым исходным кодом Zxing (Zebra Crossing) [2]. В своей работе Д.Т. Алалалыкин, П.Ю. Бугаков представили основные результаты разработки информационной системы для автоматизации учета успеваемости обучающихся. Создание информационной системы было выполнено на языке программирования Kotlin в интегрированной среде разработки Android Studio. Студенты могут просматривать свою успеваемость по дисциплинам, генерировать и предоставлять свой индивидуальный QR код для сканирования мобильным устройством преподавателя при регистрации присутствия на занятии или сдаче практической работы [3]. В своей работе А.А. Баев, А.О. Евдокимов, К.О. Костицына представили анализ системы контроля и управления доступом (СКУД) с помощью QR-кода, а также обосновали возможности её внедрения в образовательные учреждения [4].

### **1.3 Цель исследования**

Цель исследования – разработать мобильное приложение в Android Studio на языке программирования Java для генерации QR-кодов из пользовательского текста.

## **2 Материалы и методы**

Процесс создания мобильного приложения проделан в среде Android Studio, с использованием языка программирования Java. Также использована библиотека ZXing для работы с QR-кодами.

### 3 Результаты и обсуждения

Первый шаг, который необходимо выполнить – добавление зависимости в проект (рис. 1). При добавлении зависимости в файл build.gradle проекта, фактически происходит сообщение системе сборки Gradle о том, что приложение зависит от библиотеки ZXing версии 3.4.0.

```
29 }
30
31 dependencies {
32
33     implementation 'com.google.zxing:core:3.4.0'
34
35     implementation libs.appcompat
36     implementation libs.material
37     implementation libs.activity
38     implementation libs.constraintlayout
```

Рисунок 1 – Добавление зависимости в файл build.gradle

Далее создадим макет интерфейса. Добавим основные компоненты приложения в файл activity\_main.xml. Всего в макете будет три элемента: ImageView – элемент для отображения сгенерированного QR-кода, Button – элемент для инициирования процесса генерации QR-кода и EditText – поле для ввода текста, который будет включен в QR-код. Настроим параметры элементов, такие как размеры, расположение и внешний вид в соответствии с дизайном (рис. 2).

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     tools:context=".MainActivity">
7
8     <Button
9         android:id="@+id/btnGenerateQR"
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="Сгенерировать QR код"
13        android:layout_centerInParent="true"/>
14
15    <ImageView
16        android:id="@+id/imgQRCode"
17        android:layout_width="wrap_content"
18        android:layout_height="281dp"
19        android:layout_above="@+id/btnGenerateQR"
20        android:layout_alignParentStart="true"
21        android:layout_alignParentTop="true"
22        android:layout_alignParentEnd="true"
23        android:layout_marginStart="45dp"
24        android:layout_marginTop="30dp"
25        android:layout_marginEnd="45dp"
26        android:layout_marginBottom="30dp" />
27
28    <EditText
29        android:id="@+id/edtInput"
30        android:layout_width="370dp"
31        android:layout_height="wrap_content"
32        android:layout_below="@+id/btnGenerateQR"
33        android:layout_centerHorizontal="true"
34        android:layout_marginTop="16dp"
35        android:hint="Введите URL или текст" />
36
37
38 </RelativeLayout>
```

Рисунок 2 – Создание главного макета

Импортируем классы и библиотеки, необходимые для работы приложения (рис. 3).

```
1 package com.example.qrcode;
2
3 import android.graphics.Bitmap;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.ImageView;
9 import androidx.appcompat.app.AppCompatActivity;
10 import com.google.zxing.BarcodeFormat;
11 import com.google.zxing.WriterException;
12 import com.google.zxing.common.BitMatrix;
13 import com.google.zxing.qrcode.QRCodeWriter;
```

Рисунок 3 – Импорт библиотек

Объявим переменные btnGenerateQR, imgQRCode и edtInput для кнопки, изображения QR-кода и поля ввода соответственно и инициализируем их. Эти

переменные будут использоваться для взаимодействия с соответствующими элементами пользовательского интерфейса в макете. Метод `onCreate` вызывается системой при создании активности и выполняет необходимые действия для инициализации пользовательского интерфейса и установки слушателей событий. В данном случае, при нажатии на кнопку "Сгенерировать QR код", текст из поля ввода проверяется на наличие и, если он не пуст, вызывается метод для генерации и отображения, соответствующего QR-кода (рис. 4).

```
15 <> public class MainActivity extends AppCompatActivity {
16
17     private Button btnGenerateQR;
18     private ImageView imgQRCode;
19     private EditText edtInput;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25
26         btnGenerateQR = findViewById(R.id.btnGenerateQR);
27         imgQRCode = findViewById(R.id.imgQRCode);
28         edtInput = findViewById(R.id.edtInput);
29
30         btnGenerateQR.setOnClickListener(new View.OnClickListener() {
31             @Override
32             public void onClick(View view) {
33                 String inputText = edtInput.getText().toString().trim();
34                 if (!inputText.isEmpty()) {
35                     generateQRCode(inputText);
36                 }
37             }
38         });
39     }
```

Рисунок 4 – Создание метода «onCreate»

Далее пропишем метод `generateQRCode`, который отвечает за генерацию QR-кода на основе переданных данных. Этот метод использует библиотеку ZXing для создания QR-кода. Сначала он создает объект `QRCodeWriter`, затем генерирует матрицу битов для QR-кода с использованием переданных данных. Затем создается изображение (`Bitmap`), которое заполняется черными и белыми пикселями в соответствии с матрицей битов. Наконец, сгенерированный QR-код отображается в элементе `ImageView` (`imgQRCode`). Если произойдет ошибка в процессе генерации, исключение `WriterException` будет обработано, и будет выведено в консоль.

```
1 usage
50 private void generateQRCode(String data) {
51     QRCodeWriter writer = new QRCodeWriter();
52     try {
53         BitMatrix bitMatrix = writer.encode(data, BarcodeFormat.QR_CODE, width: 512, height: 512);
54         int width = bitMatrix.getWidth();
55         int height = bitMatrix.getHeight();
56         Bitmap bmp = Bitmap.createBitmap(width, height, Bitmap.Config.RGB_565);
57
58         for (int x = 0; x < width; x++) {
59             for (int y = 0; y < height; y++) {
60                 bmp.setPixel(x, y, bitMatrix.get(x, y) ? 0xFF000000 : 0xFFFFFFFF);
61             }
62         }
63
64         imgQRCode.setImageBitmap(bmp);
65
66     } catch (WriterException e) {
67         e.printStackTrace();
68     }
69 }
70 }
71
72 }
```

Рисунок 5 – Создание метода «generateQRCode»

Тестируем приложение. Для теста введем ссылку на любой сайт (рис. 6, 7).



Рисунок 6 – Интерфейс приложения



Рисунок 7 – Генерация QR кода

### **Выводы**

В данной работе был представлен пример разработки мобильного приложения для генерации QR кодов с использованием Android Studio и библиотеки ZXing. В процессе создания приложения были рассмотрены шаги по созданию пользовательского интерфейса, интеграции библиотеки ZXing для генерации QR кодов, и обработке пользовательского ввода. Результатом является функциональное приложение, способное генерировать QR коды на основе текстовых данных, введенных пользователем.

### **Библиографический список**

1. Кожевникова А.С. Разработка приложения для создания и считывания Qr-code на языке программирования java. // XVII Неделя науки молодежи СВАО. Сборник статей по итогам работы научных конференций и круглых столов. Москва, 2022. С. 404-407.
2. Халилова З.Э. Особенности создания и распознавания QR-кода в android-приложениях. // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. № 4 (22). С. 45-52.
3. Алалалыкин Д.Т., Бугаков П.Ю. Разработка информационной системы для автоматизации учета успеваемости обучающихся. // Интерэкспо Гео-Сибирь. 2023. Т. 7. № 1. С. 130-137.
4. Баев А.А., Евдокимов А.О., Костицына К.О. Анализ работы системы контроля и управления доступом (скуд) с помощью qr-кода. // Актуальные исследования. 2022. № 5 (84). С. 13-16.