УДК 004

Получение случайного изображения для мобильного приложения в Android Studio

Андриенко Иван Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье описывается процесс создания приложения для получения случайного изображения с сайта на языке программирования Java. Для получения изображения использовался метод получения данных через URL. В результате было создано приложение, генерирующее случайные изображения из интернета.
Ключевые слова: Android-разработка, мобильное приложение, изображение, Android Studio, URL.

Getting a random image for a mobile app in Android Studio

Andrienko Ivan Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article describes the process of creating an application to get a random image from a website in the Java programming language. To get the image, the method of obtaining data via URL was used. As a result, an application was created that generates random images from the Internet.
Keywords: Android development, mobile application, image, Android Studio, URL.

1 Введение

1.1 Актуальность

Большое количество пользователей используют на смартфонах функцию для автоматической смены обоев. Проблема в том, что по стандарту пользователь получает набор изображений, который при длительном использовании начинает повторяться. Со временем даже любимое изображение на обоях рано или поздно приедаются. Большинство пользователей не готовы вручную пополнять список изображений. На помощь приходит метод получения изображений из сети, с огромным выбором изображений.

1.2 Обзор исследований

В своей работе Е.Н. Нагибин выявил оптимальный путь для хранения и загрузки изображений в бизнес-приложениях на базе ОС Android [1]. А.О. Артемов, А.Р. Гончаренко сделали обзор типичных проблем при работе с изображениями на платформе ОС Android. Описано нахождение пути решения этих проблем и реализация решения типичных проблем в библиотеке Glide [2]. В своей работе М.В. Яшина, А.С. Доткулова, И.А. Кутейников представили описания алгоритмов обработки изображений и методики их реализации на языке Java в Android Studio. Представлены этапы разработки Android приложения, реализующего конкретный алгоритм обработки. Описаны основные требования, предъявляемые к обрабатываемым изображениям. [3]. Е.О. Агафонова в своей работе описала главные особенности разработки в интегрированной среде Android Studio [4].

1.3 Цель исследования

Цель исследования – создать приложение, получающее случайное изображения по нажатию на кнопку.

2 Материалы и методы

Процесс создания мобильного приложения проделан в среде Android Studio, с использованием языка программирования Java. Случайные изображения берутся с сайта.

3 Результаты и обсуждения

Главная особенность разрабатываемого приложения, автоматизация. Для получения изображения пользователь должен только нажать на кнопку. Для этого необходимо использовать сайт, который будет предоставлять случайное изображение. В данном случае использовался imgix [6].

При разработке будет использоваться интернет, поэтому необходимо получить разрешение на доступ к интернету. В файле AndoridManifest получаем его с помощью «uses permission» (рис.1).

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.lab72">
4
5      <uses-permission android:name="android.permission.INTERNET" />
6
7      <application
8          android:allowBackup="true"
```

Рисунок 1 – Получение разрешения на использование интернета

Переходим к интерфейсу приложения. В разрабатываемом приложении необходима кнопка для загрузки изображения и область для его вывода. Добавляем в макет «Button» и «ImageView» (рис. 2). Для ImageView передаем атрибут «match constraint» для заполнения всей оставшейся части экрана. Также для ImageView в атрибут srcCompat рекомендуется передать

подходящее изображение. Это покажет пользователю, где будет размещено изображение после его загрузки (рис. 3).

```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">

    <Button
        android:id="@+id/buttonDownload"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="@string/button_download"
        android:onClick="onClickDownloadImage"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonDownload"
        app:srcCompat="@android:drawable/ic_menu_gallery" />

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

Рисунок 2 – Код макета

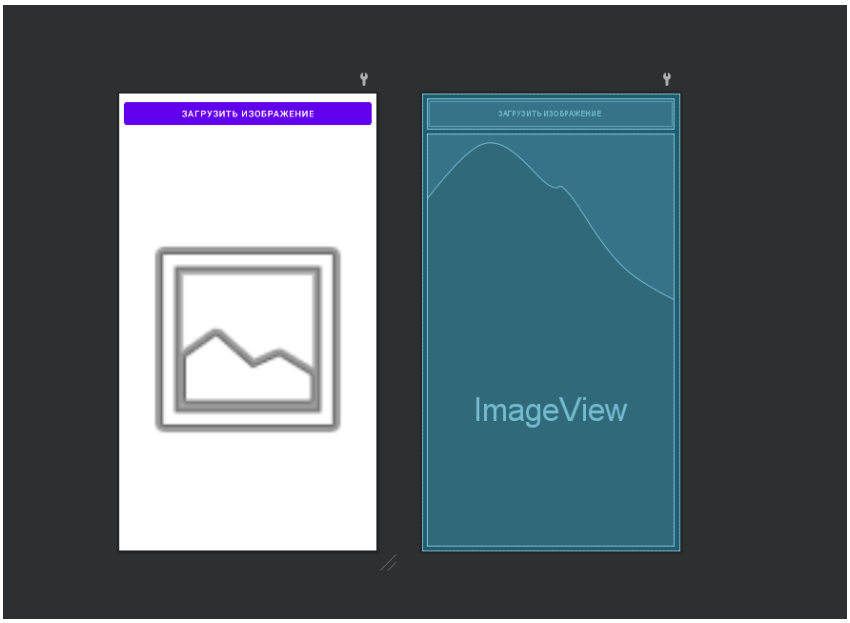

Рисунок 3 – Интерфейс приложения

В MainActivity пропишем код приложения. Так как приложение будет работать с растровой графикой импортируем необходимые для этого библиотеки, а также базовые библиотеки (рис. 4).

```
package com.example.lab72;


import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.concurrent.ExecutionException;

```

Рисунок 4 – Импорт необходимых библиотек

Далее необходимо создать переменную для ссылки на ImageView и присвоим её значение в методе onCreate с помощью «findViewById». Далее вставляем ранее найденную ссылку в строковую переменную URL (рис. 5).

```
public class MainActivity extends AppCompatActivity {

    private ImageView imageView;

    private String url = "https://source.unsplash.com/random/200x200?sig=1";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageView = findViewById(R.id.imageView);
    }
```

Рисунок 5 – Создание необходимых переменных

Теперь что бы загрузить данные из интернета создадим новый класс «DownloadImageTask», принимающий в качестве параметра строку и возвращающий изображение. В нем необходимо создать URLconnection. Сразу создадим try/catch/finally. Открываем соединение с помощью «openConnection» и создаем поток ввода. «decodeStream» декодирует входной

поток в растровое изображение. В конце try возвращаем Bitmap, а в finally закрываем соединение (рис. 6).

```
46      private static class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {
47          @Override
48          protected Bitmap doInBackground(String... strings) {
49              URL url = null;
50              HttpURLConnection urlConnection = null;
51              try {
52                  url = new URL(strings[0]);
53                  urlConnection = (HttpURLConnection) url.openConnection();
54                  InputStream inputStream = urlConnection.getInputStream();
55                  Bitmap bitmap = BitmapFactory.decodeStream(inputStream);
56                  return bitmap;
57              } catch (MalformedURLException e) {
58                  e.printStackTrace();
59              } catch (IOException e) {
60                  e.printStackTrace();
61              } finally {
62                  if (urlConnection != null) {
63                      urlConnection.disconnect();
64                  }
65              }
66              return null;
67          }
```

Рисунок 6 – Создание класса для загрузки изображения

Далее в новом классе «onClickDownloadImage» создаем объект ранее созданного класса. В переменную для изображения помещаем URL и получаем данные. Создаем исключения. В конце пишем код для установки загруженного изображение в ImageView (рис.7).

```
32
33      public void onClickDownloadImage(View view) {
34          DownloadImageTask task = new DownloadImageTask();
35          Bitmap bitmap = null;
36          try {
37              bitmap = task.execute(url).get();
38          } catch (ExecutionException e) {
39              e.printStackTrace();
40          } catch (InterruptedException e) {
41              e.printStackTrace();
42          }
43          imageView.setImageBitmap(bitmap);
44      }
```

Рисунок 7 – Создание класса для демонстрации изображения

Собираем проект и тестируем приложение. Каждый раз при нажатии на кнопку пользователь получает случайно изображение из интернета (рис. 8).

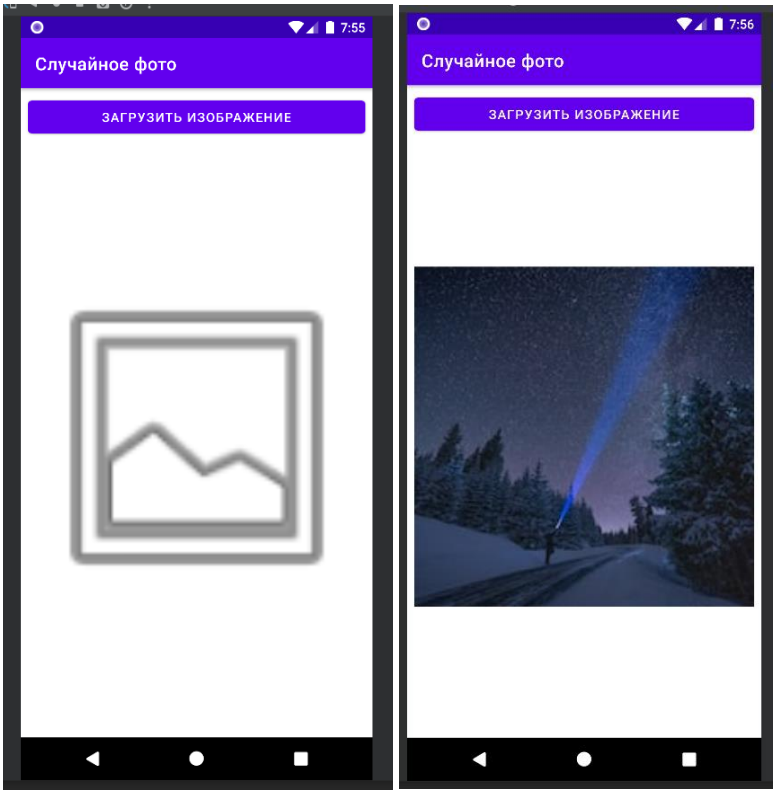

Рисунок 8 – Работа приложения

Выводы

В данной работе был описан процесс создания приложения для получения случайных изображений из интернета на языке программирования Java.

Библиографический список

1. Нагибин Е.Н. Работа с изображениями в бизнес-приложениях на базе Ос Android. // International Conference on Science, Agriculture, Engineering and Management. Conference Proceedings. 2017. С. 63-70.
2. Артемов А.О., Гончаренко А.Р. Принцип работы библиотек для загрузки изображения на ос android на примере glide. // Актуальные научные исследования в современном мире. 2018. № 5-9 (37). С. 65-68.
3. Яшина М.В., Доткулова А.С., Кутейников И.А. Методические указания по программированию алгоритмов обработки изображений для мобильных устройств на базе операционной системы ANDROID. // Москва, 2019.
4. Агафонова Е.О. Особенности работы в android studio // Вклад молодых ученых в аграрную науку. 2021. №1. С. 568-572.
5. Introduction to animations | Android Developers URL: https://developer.android.com/develop/ui/views/animations/overview
6. imgix - Image Processing On-Demand, Served by CDN URL: https://images.unsplash.com/