

# ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ РАБОТЫ С ИЗОБРАЖЕНИЯМИ В БИЗНЕС-ПРИЛОЖЕНИЯХ НА БАЗЕ ОС ANDROID

Нагибин Е.Н.

*Нагибин Евгений Николаевич – бакалавр технических наук, студент,  
кафедра инженерной геометрии и САПР, факультет элитного образования и магистратуры,  
Омский государственный технический университет, г. Омск*

**Аннотация:** темой данной статьи является выявление оптимального пути для хранения и загрузки изображений в бизнес-приложениях на базе ОС Android. Большинство Android-приложений в качестве источника изображений используют Интернет. Это продиктовано как тем, что большинство продуктов на Android используют Интернет-соединение, тем самым делая отображаемую в них информацию уникальной и не поддающейся предварительной загрузке, так и тем, что изображения, содержащиеся в исполняемом файле apk (формат архивных файлов-приложений в Android) увеличивают размер этого файла, что усложняет его скачивание из Google Play (магазина приложений в Android). В ходе данной статьи в качестве бизнес-приложения будет рассматриваться экран со списком новостей, в котором каждая новость будет уникальной и состоять из изображения, заголовка и описания.

*Для выявления оптимального пути хранения и загрузки изображений необходимо:*

- 1) Провести обзор существующих способов работы с изображениями в Android*
- 2) Провести испытания всех найденных способов в равных условиях*
- 3) Выбрать наиболее подходящий путь для хранения и загрузки приложений*

**Ключевые слова:** *Android, кэширование изображений.*

УДК 612.822.1+612.45.015.3

## **Введение**

Хранение и загрузка изображений является неотъемлемой частью абсолютного большинства приложений на Android. Зачастую показываемые изображения в данных продуктах имеют различные размеры и качество. В качестве примера возьмем новостное приложение. В нем на главном экране содержится список с новостями (рис. 1), представленными в виде заголовка, описания и непосредственно изображения. Данные изображения приходят из сети в различном формате. В приложении существуют два вида новостей, которые различаются по важности содержащейся в них информации, таким образом, у главных новостей изображение показывается в большом размере и показывается посередине, создавая акцент именно на этой новости. В новостях обычного типа картинка имеет небольшой размер и ставится в правом верхнем углу. Пользователь часто не акцентирует внимание на таких изображениях.

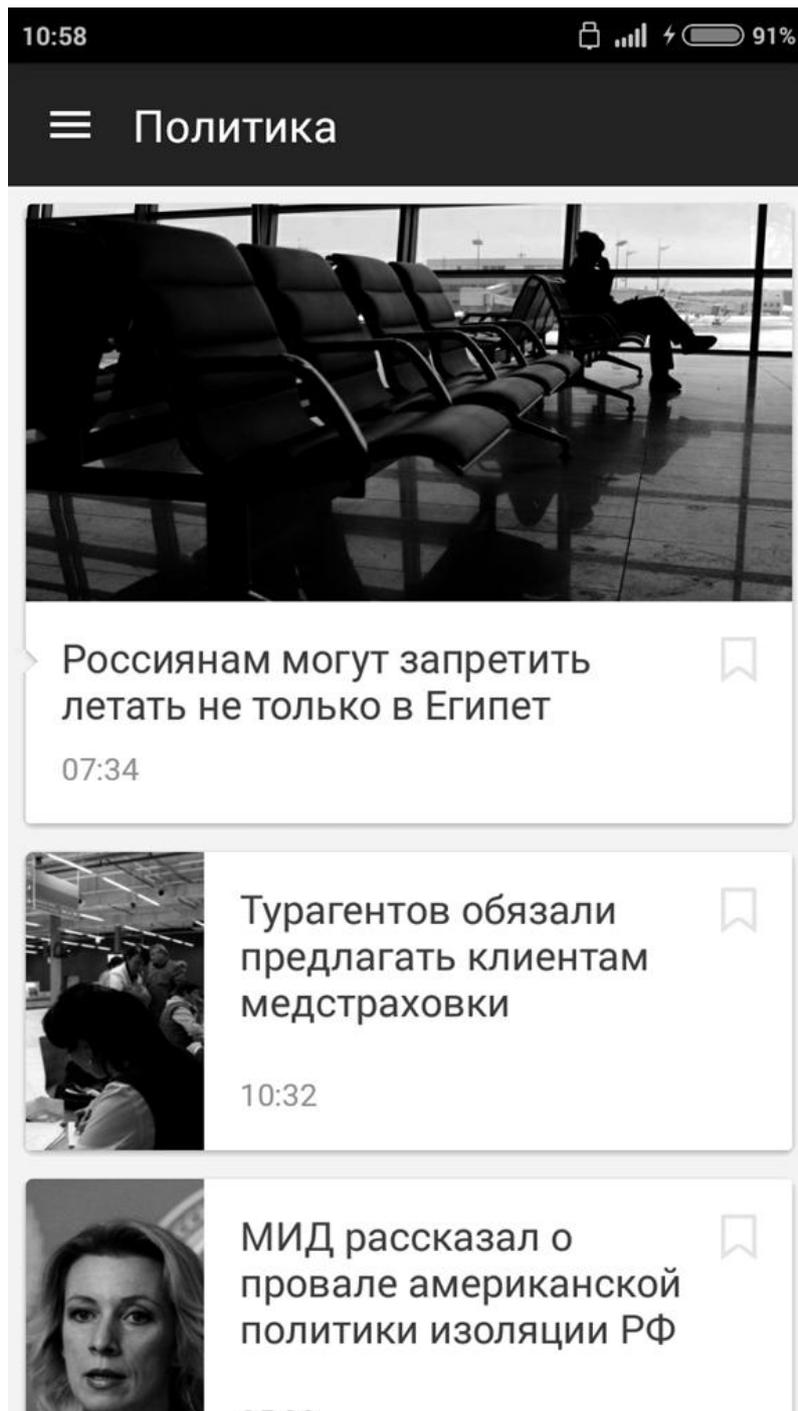


Рис. 1. Новостное приложение

### Постановка задачи

Одной из важнейших проблем, стоящей в вопросе загрузки и хранения изображений является их кэширование, поскольку пользователь может иметь, как слабое Интернет-соединение, так и ограниченный или дорогой трафик, поэтому необходимо обработать ситуацию таким образом, чтобы загруженное единожды изображение сохранялось в память устройства (либо оперативную, либо на диске) и при следующем запросе данного изображения, оно уж загружалось не из сети, а из хранилища.

В вопросе кэширования необходимо определить, какие картинки сохранять. Как упоминалось ранее, изображения в Android могут иметь различный формат, но в то же время повторяться. Если взять в качестве примера новостное приложение, то одно и то же изображение может показываться в различном формате на разных экранах, но при этом иметь одну и ту же ссылку. Это продиктовано дизайном приложения. Соответственно, при каждой загрузке изображения оно обрезается или масштабируется под размер отведенного на экране места, что занимает время при отрисовывании изображения. Таким

образом, возникает дилемма, какие изображения хранить в памяти устройства при кэшировании: измененные или оригиналы.

По умолчанию, Android не предоставляет способа работы с загрузкой/хранением изображений, поэтому разработчик вынужден использовать сторонние библиотеки, либо создавать свои для оптимальной работы с изображениями.

Для выявления оптимального пути хранения и загрузки изображений необходимо:

- 1) Провести обзор существующих способов работы с изображениями в Android.
- 2) Провести испытания всех найденных способов в равных условиях.
- 3) Выбрать наиболее подходящий путь для хранения и загрузки приложений.

#### **Теория**

В Android существует множество сторонних библиотек для работы с приложениями, будет проведен обзор самых популярных из них: Picasso, Glide и Universal Image Loader (UIL). Все библиотеки имеют общие цели и схожую структуру, остановимся на каждой из них подробнее[2].

Picasso – библиотека, названная именем известного художника Пабло Пикассо и созданная Джеком Уортоном; она предназначена для асинхронной загрузки изображений из сети, ресурсов или файловой системы, их кэширования и отображения.

Glide – фреймворк для работы с медиа, который декодирует файлы и кэширует их Создан компанией Vump Technologies [1].

Universal Image Loader (UIL) – инструмент для асинхронной загрузки изображений, их кэширования, редактирования и отображения[4].

В основном, функции данных библиотек одинаковы и они имеют схожую структуру. Выделим наиболее важные различия:

1) Picasso хранит преобразованные изображения в памяти, UIL – оригиналы. В Glide можно определить, какой тип сохранять[3].

2) Glide – единственная из трех библиотек, способная поддерживать gif-изображения, также бонусом она может работать и с видео.

3) Glide также предотвращает возникновение OutOfMemoryError – самой популярной ошибки при работе с изображениями в Android, возникающей, когда размер загружаемого изображения превышает выделенную под него память устройства.

#### **Результаты экспериментов**

Сравнение Glide, Picasso и Universal Image Loader будет идти по следующим параметрам [3]:

1) Размер библиотеки — ранее уже упоминалось, что чем меньше исполняемый файл арк, тем лучше (рис. 2).

2) Количество методов в библиотеке. Максимальное количество методов(функций) в приложении составляет 65536, превысить этот лимит можно, но это усложнит разработку(рис. 3).

3) Потребляемая память (рис. 4).

4) Работа с сетью и загрузка ЦПУ.

Сравнение было проведено при помощи Android Studio – основной IDE для разработки под Android. Список новостей из двухсот уникальных новостей различного типа.

Полученные результаты:

<b>Size in Kb</b>	
<b>Library</b>	<b>Size</b>
<b>Picasso</b>	<b>118kb</b>
<b>Universal Image Loader</b>	<b>157kb</b>
<b>Glide</b>	<b>464kb</b>

*Рис. 2. Размер библиотек в килобайтах*

<b>Amount of Methods</b>	
<b>Library</b>	<b>Methods</b>
Picasso	446
Universal Image Loader	865
Glide	2363

Рис. 3. Количество методов

<b>Memory Consumption</b>		
<b>Library</b>	<b>Min Memory</b>	<b>Max Memory</b>
Picasso (default mem cache limit)	117mb	130mb
Picasso (with 25MB mem cache limit)	50mb	65mb
Universal Image Loader	45mb	55mb
Glide	55mb	65mb

Рис. 4. Потребляемая память

#### Обработка результатов

Как видно из рисунков, Picasso выигрывает по количеству методов и общему размеру библиотеки. Но в силу этого, функционал данной библиотеки проигрывает остальным. По потреблению памяти с небольшим преимуществом выиграл Universal Image Loader, но при этом у Picasso необходимо было указать лимит кэширования в 25 Мб, что сильно ограничивает количество содержащихся в хранилище изображений.

Работа с ЦПУ и сетью у библиотек построена схоже, тем самым, различий замечено не было.

Что касается сложности интеграции данных библиотек в приложение, то UIL потребует больше строк кода и более скрупулезную конфигурацию. Glide и Picasso имеют очень схожий синтаксис, но Glide прекрасно работает «из коробки», а Picasso потребует конфигурированию (в частности, размер кэша для оптимального потребления памяти). Также у Picasso были замечены утечки памяти в ситуациях, в которых у других библиотек не возникло таких проблем.

#### Выводы и заключение

В данной статье были рассмотрены и исследованы три наиболее популярных библиотеки под Android Picasso, Glide и Universal Image Loader прекрасно подходят для решения своих задач и у каждой из них есть свои преимущества перед другими, но в качестве библиотеки для работы с изображениями при недостатке времени на разработку собственного решения, я бы использовал Glide, несмотря на то, что размер данного фреймворка превышает размер конкурентов в более, чем в 2 раза, поскольку он очень прост в интеграции, поддерживает gif-изображения, а также рекомендован Google для разработки под Android.

Если же есть достаточно времени для разработки своего решения, то я бы рекомендовал разработать собственную библиотеку, взяв за основу Glide и усовершенствовав под собственные нужды, например, добавление сжатия изображений для оптимального кэширования, сокращение размера библиотеки путем удаления неиспользуемого функционала и т.д.

#### Список литературы

1. Glide. [Электронный ресурс]: Электрон. дан. [Б. м.]. Режим доступа: <https://github.com/bumptech/glide/> (дата обращения: 13.12.2016).
2. Picasso, Universal Image Loader or Glide. [Электронный ресурс]: Электрон. дан. [Б. м.]. Режим доступа: <https://hackernoon.com/picasso-universal-image-loader-or-glide-that-s-the-question/> (дата обращения: 13.12.2016).
3. Picasso – Powerful image downloading. [Электронный ресурс]: Электрон. дан. Режим доступа: <http://square.github.io/picasso/> (дата обращения: 14.12.2016).
4. Universal Image Loader. [Электронный ресурс]: Электрон. дан. Режим доступа: <https://github.com/nostr13/Android-Universal-Image-Loader/> (дата обращения: 14.12.2016).