**ВВЕДЕНИЕ**

По данных Counterpoint Research, более половины пользователей проводят больше пяти часов в сутки за смартфоном. Можно сказать, что приложения для мобильных платформ актуальны теперь как никогда раньше.

Сегодня абсолютное большинство людей имеют аккаунты в нескольких социальных сетях. Twitter – популярная социальная сеть, характерной особенностью которой является особый стиль записей («твитов», от англ. «tweet» – щебетать). Ограничение каждого «твита» в 280 символов обуславливает особый лаконичный характер публикуемых пользователями сообщений.

В Twitter можно читать политических и общественных деятелей, пользователями Twitter являются большинство крупных организаций и брендов.

Целью данной работы является разработка Twitter-клиента для ОС Android. Twitter-клиент – приложение, осуществляющее обмен информацией с сервером Twitter посредством API. Минимальная функциональность должна включать в себя возможность просматривать новостную ленту, добавлять новые «твиты», просматривать аккаунты пользователей.

Заинтересованность в создании данного программного продукта обусловлена возможностью изучить разработку ПО для платформы Android, получить представление об основах Material Design, а также получить опыт работы с сетью.

**1 ОБЗОР ИСТОЧНИКОВ**

Среди языков программирования, популярных среди Android-разработчиков выделяются языки Java и Kotlin. В качестве языка, используемого для написания приложения, выбран язык Java, так как синтаксис и некоторые средства Java изучались в курсе Кроссплатформенного Программирования, а также существует большое количество источников для изучения опыта написания программ под платформу Android.

В отличие от языка С++ (от которого произошёл язык Java), многие аспекты Java созданы для удобства разработчика. Объектно-ориентированный подход, используемый в Java, предоставляет программисту возможность писать более читаемый код, а также легко адаптироваться к специфике любой задачи.

Разметка выполнена с помощью набора инструментов XML, по умолчанию предоставляемого Android. Такой вид разметки позволяет разделять графический пользовательский интерфейс от логики программы. Таким образом, внешний вид приложения можно менять независимо от кода.

В качестве рекомендуемой среды разработки под Android Google предлагает Android Studio. Android Studio предоставляет возможности разработки, отладки и сборки многофайловых проектов, а также позволяет эмулировать конфигурации различных устройств с целью тестирования приложений.

Для взаимодействия с сервером Twitter использована Twitter SDK. Отрицательной стороной использования данной библиотеки является практически полное отсутствие структурированной документации, а также неработоспособность некоторых модулей. Положительной – существующие готовые решения типовых проблем, возникающих в ходе разработки клиентских приложений для Twitter.

Дополнительно для асинхронной загрузки изображений из сети была использована библиотека Picasso, также используемая авторами Twitter SDK.

**2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ**

Наиболее существенная задача приложения-клиента – организовать коммуникацию между клиентм и сервером, а также взаимодействие пользователя с клиентом (косвенно, с сервером). Соответственно, приложение должно предоставлять удобный интерфейс пользователя, в зависимости от состояния которого вызываются необходимые участки кода и выполняется логика программы.

В зависимости от манипуляций пользователя, механизм клиента отправляет запросы серверу на получение некоторой информации и\или запрашивает изменения данных, хранящихся на сервере. Затем клиент ожидает ответа, обрабатывает его и меняет собственные данные, которые в изменённом виде необходимо отобразить в интерфейсе.

Графический интерфейс пользователя данного приложения объединяет несколько xml-файлов, соответствующих главному экрану, странице информации о пользователе, страницу поиска и различные вспомогательные элементы: фрагменты ленты твитов, меню тулбара и др.

Такие затратные с точки зрения процессорного времени операции, как, например, сетевые запросы, целесообразно выполнять асинхронно по отношению к основному потоку. В качестве инструмента для решения этой задачи выбран AsyncTask.



**3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

В объектно-ориентированном подходе к программированию функциональным блокам соответствуют классы. Рассмотрим наиболее существенные классы. Диаграмма классов продемонстрирована в приложении Б.

**3.1 Класс LoginActivity**

Класс, реализующий авторизацию и хранение информации о ней. Страница, соответствующая этому классу, не отображается, если пользователь уже авторизован.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| loginButton | TwitterLoginButton | Механизм авторизации из Twitter SDK | |
| token | String | Информация о сессии | |
| secret | String |
| session | TwitterSession | Актуальная Twitter-сессия | |
| isUserAuthorized | Boolean | Информация о том, авторизован ли пользователь | |
| mShared | SharedPreferences | Механизм постоянного хранения информации об авторизации | |
| Private методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| saveSessionInfo | String, String | Void | метод, используемый для записи информации о сессии в SharedPreferences |
| getSessionInfo | Void | Void | метод, используемый для получения информации о сессии из SharedPreferences |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity в Android, составляющая паттерна «Шаблонный метод» |
| onActivityResult | Int, int, Intent | Void | Передаёт результат авторизации в LoginButton |

**3.2 Класс TimelineActivity**

Данный абстрактный класс соответствует любому Activity-классу, содержащему в себе ленту «твитов».

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| tweetsRecyclerView | RecyclerView | Элемент View, используемый для отображения «твитов» | |
| tweetAdapter | TweetAdapter | Объект, формирующий фрагмент RecyclerView из информации о «твите» | |
| task | AsyncTask<Void, Void, Void> | Средство асинхронного выполнения запроса к серверу | |
| Package-private методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| initToolbar | Void | Void | Метод инициализации toolbar |
| initTask | Void | Void | Метод инициализации AsyncTask. Абстрактен, т.к. каждый класс Activity отображает особый вид ленты, а значит, информация о «твитах» запрашивается по-разному |
| loadTweets | Void | Void | Загружает «твиты» |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity |

**3.3 Класс StandardTimelineActivity**

Наследник TimelineActivity, особенность которого – стандартный тулбар с меню приложения.

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreateOptionsMenu | Menu | Boolean | Метод инициализации меню в тулбаре |
| onOptionsItemSelected | MenuItem | Boolean | Метод обработки нажатия на иконки в тулбаре |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity. Инициализирует все поля и вызывает метод loadTweets() родительского класса |

**3.4 Класс MainActivity**

MainActivity – «минимальный» класс-наследник StandardTimelineActivity, переопределяет методы OnCreate и initTask (инициализирует поле task родительского класса через TimelineTask).

**3.5 Класс UserInfoActivity**

Класс – клас-наследник StandartTimelineActivity, имеющий, соответственно, тулбар с меню, RecyclerView для отображения твитов. Так как класс представляет собой логику страницы информации о пользователе, особенность класса – наличие блока набора View для отображения основных данных о пользователе.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| userId | Long | ID пользователя | |
| nameTextView, nickTextView,  userImageView,  descriptionTextView,  followingCountTextView,  followersCountTextView | View | Объекты блока данных о пользователи | |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| displayUserInfo | User | Void | Заполняет поля информации о пользователе актуальной информацией |
| Package-private методы класса | | | |
| initTask | Void | Void | Метод инициализации AsyncTask (UserInfoTask) |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity, инициализирует все элементы интерфейса по значению ID пользователя, получаемого из extras |

**3.6 Класс SearchActivity**

Данный класс реализует поиск «твитов» по запросу пользователя.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| searchText | EditText | Элемент View, используемый для ввода пользователем поискового запроса | |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| initToolBar | Void | Void | Метод инициализации поля ввода запроса в тулбаре |
| initTask | Void | Void | Метод инициализации AsyncTask (SearchTask) |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity. |

**3.7 Класс TweetAdapter**

Данный класс является адаптером, предназначенных для создания специальных элементов, отображающих информацию о «твите», и заполнения ими RecyclerView. TweetAdapter имеет в себе вложенный класс TweetViewHolder, реализующий паттерн ViewHolder, суть которого в создании объекта, хранящего ссылки на все элементы View, для каждого фрагмента RecyclerView с целью минимизировать количество дорогостоящего в смысле затрат процессорного времени метода findViewById.

**4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

**5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**6 ТЕСТИРОВАНИЕ**