ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Преподаватель кафедры УРПО отделения  Программной инженерии  Факультета Бизнес-информатики | УТВЕРЖДАЮ  заведующий отделением  Программной инженерии,  факультета Бизнес-информатики,  профессор кафедры УРПО |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пронин С.А. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Авдошин С.М. |
| “\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_2014 г. | “\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_2014 г. |
|  |  |

**ANDROID ПРИЛОЖЕНИЕ С ДИНАМИЧЕСКИ УПРАВЛЯЕМОЙ СТРУКТУРОЙ И СОДЕРЖАНИЕМ КОНТЕНТА.**

**Пояснительная записка**

Инв. № подл.

Подпись и дата

Взам. инв. №

Инв. № дубл.

Подпись и дата

**Лист утверждения**

**А.В.00001-01 ТЗ 01**

**RU.17701729.508900-01 81 01-1**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | Исполнитель: студент группы 273ПИ |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_/Шугуров И.С./ |
|  |  | “\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2014 г. |

RU.17701729.508900-01 ТЗ 01-1

**УТВЕРЖДЕНО**

RU.17701729.508900-01 81 01-1

**ANDROID ПРИЛОЖЕНИЕ С ДИНАМИЧЕСКИ УПРАВЛЯЕМОЙ СТРУКТУРОЙ И СОДЕРЖАНИЕМ КОНТЕНТА.**

**Пояснительная записка**

Инв. № подл.

Подпись и дата

Взам. инв. №

Инв. № дубл.

Подпись и дата

**RU.17701729.508900-01 81 01-1**

RU.17701729.508900-01 ТЗ 01-1

# Содержание

[1. Введение 3](#_Toc387613652)

[2. Назначение и область применения 4](#_Toc387613655)

[2.1 Назначение разработки 4](#_Toc387613656)

[2.2 Краткая характеристика области применения программы 4](#_Toc387613657)

[3. Технические характеристики 5](#_Toc387613658)

[3.1 Постановка задачи на разработку 5](#_Toc387613661)

[3.2 Математические определения используемых терминов 5](#_Toc387613662)

[3.3 Описание алгоритма функционирования программы 5](#_Toc387613663)

[3.4 Описание и обоснование выбора метода организации входных и выходных данных 7](#_Toc387613668)

[3.5 Описание и обоснование состава технических и программных средств 8](#_Toc387613671)

[4. Список источников, используемых при разработке 9](#_Toc387613672)

[Приложение 1 Описание и функциональное назначение классов 10](#_Toc387613673)

[Приложение 2 описание и функциональное методов, полей и свойств 11](#_Toc387613675)

[Лист регистрации изменений 17](#_Toc387613677)

# ВВЕДЕНИЕ

## Наименование программы:

Наименование программы: - «Информационное приложение отделения программной инженерии факультета бизнес-информатики национального исследовательского университета “Высшая Школа Экономики”»

Краткое наименование - «Отделение ПИ»

## Основания для разработки

Приказ № 6.18.1-02/2911-14 от 29.11.13 «Об утверждении тем и руководителей курсовых работ студентов 2 курса отделения программной инженерии факультета бизнес-информатики».

Организация, утвердившая этот документ: Национальный исследовательский университет «Высшая школа экономики», факультет Бизнес-информатики, отделение программной инженерии, кафедра управления разработкой программного обеспечения.

Наименование темы разработки: «Android приложение с динамически управляемой структурой и содержанием контента».

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## Назначение разработки

Приложение реализует клиентскую часть клиент-серверной архитектуры.

## Краткая характеристика области применения программы

Приложение будет использоваться абитуриентами и студентами отделения программной инженерии факультета бизнес-информатики национального исследовательского университета «Высшая Школа Экономики» для получения актуальной информации об отделении и университете.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ



## Постановка задачи на разработку

Разработка приложение, которое облегчит доступ к важным документам и новостям для абитуриентов и студентов отделения программной инженерии факультета бизнес-информатики национального исследовательского университета «Высшая Школа Экономики»

## Описание и обоснование выбора метода организации входных

Данные с сервера получаются в формате JSON, потому что на момент начала написания приложение данное API уже существовало и использовалось.

## Описание и обоснование состава технических и программных средств

* Приложение было написано для операционной системы Android так как это самая распространённая мобильная операционная система, а так же потому, что под Android не было аналогов;
* Языком реализации была выбрана Java, так как это стандартный язык для написания приложений под Android;
* Программа реализована в среде разработки Android Studio 0.5.8;
* Для запуска необходим Android версии 2.2 или старше, OpenGL ES version 2, Google Play Services.

# СПИСОК ИСТОЧНИКОВ, ИСПОЛЬЗУЕМЫХ ПРИ РАЗРАБОТКЕ

* Gamma, E. Design Patterns: Elements of Reusable Object-Oriented Software/ E. Gamma и др; - 1-е изд. - Addison-Wesley Professional; США, 1994 – 416с
* Freeman, E: Head First Design Patterns/ Eric Freeman и др. – 1-е изд. - O'Reilly Media, США, 2004 – 638 с.
* Lee, W: Beginning Android 4 Application Development/ Wei-Meng Lee; - 1-е изд. – Wrox, США, 2012. – 560 c.
* Komatineni, S: Pro Android 4/ Satya Komatineni, Dave MacLean ; - 1-е изд. – Apress, США, 2012. – 1020 c.

# ПРИЛОЖЕНИЕ 1

# ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

## Пакет ru.hse.shugurov.gui

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс** | **Суперкласс** | **Назначение** | |
| *AbstractFragment* | *Fragment* | Класс определяет базовый функционал и набор полей для отображения раздела. |
| *AuthenticationFragment* | *Fragment* | Предназначен для авторизации в социальных сетях. |
| *CommentsFragment* | SocialNetworkAbstractList | Класс предназначен для демонстрации комментариев к обсуждениям в социальных сетях. |
| *EventsFragments* | *AbstractFragment* | Класс предназначен для отображения раздела со списком событий. |
| *EventsListAdapter* | *BaseAdapter* | Класс поставляет объекты для списка событий. |
| *HTMLScreenFragment* | *AbstractFragment* | Класс предназначен для показа раздела с HTML контентом. |
| *InternalWebFragment* | *AbstractFragment* | Класс предназначен для показа веб-страницы внутри приложения. |
| *MapFragment* | *SupportMapFragment* | Класс предназначен демонстрации раздела с картой. |
| *RSSFragment* | *AbstractFragment* | Класс предназначен для демонстрации раздела с RSS элементами. |
| *RSSListAdapter* | *BaseAdapter* | Класс поставляет объекты для списка RSS элементов. |
| *ScreenFactory* | *Object*.Реализует интерфейс *Serializable* | Класс, реализующий фабричный метод. Используется при переходе из одного раздела в другой. |
| *SocialNetworkAbstractList* | *ListFragment* | Класс определяет базовый функционал и набор полей для раздела с социальной сетью. |
| *TopicCreationFragment* | *Fragment* | Класс для создания новых обсуждений в социальных сетях. |
| *TopicsFragment* | *SocialNetworkAbstractList* | Предназначен для отображения списка тем обсуждений из социальной сети. |
| *ContainerFragment* | *AbstractFragment.* Реализует интерфейс *View.OnClickListener* | Предназначен для отображения списка разделов. |
| *WallCommentsFragment* | *SocialNetworkAbstractList* | Предназначен для отображения постов со стены публичной страницы ВКонтакте. |
| *WallPostScreen* | *SocialNetworkAbstractList* | Предназначен для отображения постов со стены публичной страницы ВКонтакте и комментариев к нему. Позволяет добавлять комментарии. |
| *AccessTokenRequest* | Реализует интерфейс *Serializable* | Интерфейс для обратного вызова. Возвращает токен доступа. |

## Пакет ru.hse.shugurov.screens

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс** | **Суперкласс** | **Назначение** | |
| *BaseScreen* | *Object,* реализует интерфейс *Parcelable* | Класс определяет базовый функционал и набор полей раздела. |
| *Event* | *BaseScreen* | Описывает событие. |
| *EventScreen* | *BaseScreen,*реализует интерфейс *HasFile* | Описывает раздел со списком событий. |
| *FileDescription* | *Object* | Описывает файл как набор из ссылки на файл, названия файла и булевской переменной, указывающей на то, возможно ли изменение файла с течением времени |
| *HasFile* | *Object* | Интерфейс, который должна реализовать классы разделов, для которых необходимо скачать файл. |
| *HtmlContentScreen* | *BaseScreen,* реализует интерфейс *HasFile* | Описывает раздел с HTML контентом. |
| *MapScreen* | *BaseScreen* | Описывает раздел с картой. |
| *MarkerWrapper* | *Object,* реализует интерфейс *Parcelable* | Служит обёрткой для объектов класса *MarkerOptions,* добавляя несколько дополнительных полей: адресс, телефон, ссылка на веб-страницу. |
| *RSSScreen* | *BaseScreen* | Описывает конкретный элемент RSS |
| *RSSTypes* | Реализует интерфейс *Serializable* | Перечисление, содержащее 2 типа RSS элементов: с текстом и без. |
| *RSSWrapperScreen* | *BaseScreen,* реализует интерфейс *HasFile* | Описывает раздел со списков RSS элементов |
| *ScreenTypes* | Object | Перечисляет все возможные типы разделов и их код в API серверной части. |
| *ScreenWithFile* | *BaseScreen,* реализует интерфейс *HasFile* | Используется для описания раздела, который должен открывать файл во внешнем приложении. |
| *SocialNetworkScreen* | BaseScreen | Определяет набор полей для раздела с социальной сетью. |

**Пакет ru.hse.shugurov.** **social\_networks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс** | **Суперкласс** | **Назначение** | |
| *AbstractRequester* | *Object,* реализует интерфейс *Serializable* | Абстрактный класс, который описывает интерфейс для классов, реализующих запросы к социальным сетям. Предоставляет пустые реализации все методом, таким образом производные классы переопределяют только те методы, которые имею смысл для конкретной социальной сети |
| *AccessToken* | *Object,* реализует интерфейс *Serializable* | Класс описывает токен доступа для социальной сети. |
| *AuthorizationWebClient* | *WebViewClient* | Переопределённый метод shouldOverrideUrlLoading этого класса следит за переадресацией запроса, ожидая запрос, который начнётся с определённой подстроки. Когда такой запрос получен, то извлекает из него токен доступа и время действия токена. |
| *CommentsAdapter* | *BaseAdapter* | Класс поставляет объекты для списка комментариев. |
| *FacebookRequester* | *AbstractRequester* | Класс предоставляет функционал для формирования и осуществления запросов к Facebook. |
| *SocialNetworkEntry* | *Object,* реализует интерфейс *Parcelable* | Описывает поля, которые присутствуют во всех записях социальных сетей: автора записи, текст записи и дату записи. |
| *SocialNetworkProfile* | *Object,* реализует интерфейс *Parcelable* | Описывает профиль в социальной сети. |
| *SocialNetworkTopic* | *SocialNetworkEntry* | Описывает тему обсуждений в социальной сети. Помимо полей, наследованных из суперкласса, предоставляет следующие поля: заголовок, уникальный идентификатор, количество комментариев, прикреплённую картинку. |
| *StateListener* | *Object,* реализует интерфейс *Serializable* | Объекты этого класса служат для инкапсуляции оповещения об изменении состояния. |
| *TopicsAdapter* | *BaseAdapter* | Класс поставляет объекты для списка обсуждений. |
| *VKRequester* | *AbstractRequester* | Класс предоставляет функционал для формирования и осуществления запросов к ВКонтакте. |
| *VkWallPostsAdapter* | *BaseAdapter* | Класс поставляет объекты для списка записей на стене ВКонтакте. |

**Пакет ru.hse.shugurov.** **utils**

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс** | **Суперкласс** | **Назначение** | |
| *Downloader* | *AsyncTask<FileDescription, Void, Void>* | Служит для скачивания файлов в отдельном потоке |
| *FileManager* | *Object* | Предназначен для удобного считывания файла |
| *ImageLoader* | *Object* | Предназначен для кэширования изображений и для загрузки новых. Использует класс LruCache для кэширования изображений. |
| *ImageViewProxy* | *ImageView* | Предназначен для корректного отображения изображений с неизвестным соотношением сторон. |
| *Requester* | *AsyncTask<String, Void, String[]>* | Осуществляет запросы к социальным сетям. |

# ПРИЛОЖЕНИЕ 2

# ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ И ПОЛЕЙ

*AbstractFragment*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| setArguments | public | void | Bundle args | Задаёт аргументы для отображения фрагмента |
| readStateFromBundle | private | void | Bundle args | Считывает аргумент из Bumdle объекта, инициализирует поля класса |
| onCreate | public | void | Bundle savedInstanceState | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. |
| configureActionBarn | private | void | - | Показывает заголовок в ActionBar, при необходимости добавляет туда кнопки |
| onResume | public | void | - | Запрашивает конфигурацию ActionBar |
| onSaveInstanceState | public | void | Bundle outState | Сохраняет состояние фрагмента |
| onCreateOptionsMenu | public | void | Menu menu, MenuInflater inflater | Устанавливает меню в ActionBar |
| getScreen | protected | BaseScreen | - | Возвращает описание фрагмента, установленное в поле screen. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Назначение** |
| SCREEN\_TAG | public | String | Используется как ключ в Bundle объекте для передачи состояния фрагмента. |
| screen | private | BaseScreen | Объект, который описывает, что необходимо отображать фрагменту. |

*AuthenticationFragment*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | |
| **Имя** | **Мод. доступа** | | **Тип** | **Аргументы** | **Назначение** | |
| setArguments | public | | void | Bundle args | Задаёт аргументы для отображения фрагмента | |
| readStateFromBundle | private | | void | Bundle args | Считывает аргумент из Bumdle объекта, инициализирует поля класса | |
| onCreate | public | | void | Bundle savedInstanceState | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. | |
| onCreateView | public | | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | Создаёт элементы пользовательского интерфейса. | |
| onSaveInstanceState | public | | void | Bundle outState | Сохраняет состояние фрагмента | |
| createWebViewClient | private | | WebClient | String url | Создаёт WebClient для заданной социальной | |
| **Поля** | | | | | | |
| **Имя** | | **Мод. доступа** | | **Тип** | | **Назначение** |
| URL\_TAG | | public | | String | | Используется как ключ в Bundle объекте для передачи ссылки для авторизации. |
| TOKEN\_REQUEST\_TAG | | public | | String | | Используется как ключ в Bundle объекте для хранения объекта для обратного вызова. |
| url | | private | | String | | Ссылка для авторизации. |
| accessTokenRequest | | private | | AccessTokenRequest | | Объект для обратного вызова. |

*Interface AccessTokenRequest*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| receiveToken | public | void | AccessToken accessToken | Оповещает о получении токена. |

*CommentsFragment*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | |
| **Имя** | **Мод. доступа** | | **Тип** | | **Аргументы** | **Назначение** | |
| setArguments | public | | void | | Bundle args | Задаёт аргументы для отображения фрагмента | |
| getCommentsFromBundle | private | | void | | Bundle args | Извлекает комментарии из Bundle объекта. | |
| readStateFromBundle | private | | void | | Bundle args | Считывает аргумент из Bumdle объекта, инициализирует поля класса | |
| onCreate | public | | void | | Bundle savedInstanceState | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. | |
| onViewCreated | public | | void | | View view, Bundle savedInstanceState | Создаёт элементы пользовательского интерфейса. | |
| loadComments | private | | void | | - | Запрашивает загрузку комментариев. | |
| setAdapter | private | | void | | - | Устанавливает адаптер для ListView/ | |
| createHeaderView | private | | void | | - | Создаёт элемент, который показывается вверху списка. | |
| createFooterView | private | | void | | - | Создаёт элемент, который показывается внизу списка. | |
| onSaveInstanceState | public | | void | | Bundle outState | Сохраняет состояние фрагмента | |
| **Поля** | | | | | | | |
| **Имя** | | **Мод. доступа** | | **Тип** | | | **Назначение** |
| TOPIC\_ID\_TAG | | public | | String | | | Используется как ключ в Bundle объекте для передачи идентификатора обсуждения. |
| COMMENTS\_TAG | | public | | String | | | Используется как ключ в Bundle объекте для передачи массива комментариев. |
| COMMENTS\_LISTENER\_TAG | | public | | String | | | Используется как ключ в Bundle объекте для передачи объекта обратного вызова. |
| TOPIC\_TITLE\_TAG | | public | | String | | | Используется как ключ в Bundle объекте для передачи названия обсуждения. |
| COMMENTS\_COMMENT\_TAG | | public | | String | | | Используется как ключ в Bundle объекте для сохранения комментария, который начал вводить пользователь. |
| topicId | | private | | String | | | Хранит идентификатор обсуждения. |
| comments | | private | | SocialNetworkEntry[] | | | Хранит массив комментариев. |
| footerView | | private | | View | | | Хранит элемент внизу списка комментариев. |
| input | | private | | EditText | | | Поле для ввода комментария. |
| commentText | | private | | String | | | Текст вводимого пользователем комментария. |
| stateListener | | private | | StateListener | | | Объект для оповещения о том, что добавился комментарий. |
| topicTitle | | private | | String | | | Хранит заголовок обсуждения. |
| headerView | | private | | View | | | Хранит элемент вверху списка комментариев. |

*ContainerFragment*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | **Аргументы** | | | **Назначение** | |
| getLinearLayoutWithScreenItems | | private | LinearLayout | LayoutInflater inflater, BaseScreen[] elements, int screenWidth | | | Создаёт графическое представление списка разделов. | |
| setAppearance | | private | void | View view, int HSEViewType | | | Устанавливает цвет фона и изображение. | |
| onClick | | public | void | View view | | | Обрабатывает нажатия на экран. | |
| getScreenSizeAfterAPI13 | | private | int | Display display | | | Получает ширину экрана. | |
| getScreenSizeBeforeAPI13 | | private | int | Display display | | | Получает ширину экрана. | |
| getScreenWidth | | private | int | - | | | Рассчитывает ширину экрана | |
| onCreateView | | public | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | | | Создаёт элементы пользовательского интерфейса. | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | **Тип** | **Назначение** | |
| MINIMUM\_WIDTH\_OF\_THE\_ELEMENT | private | | | | int | Хранит минимальную ширину элемента. | |

*EventsFragment*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| onCreateView | public | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | Создаёт элементы пользовательского интерфейса. |

*EventsListAdapter*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Конструкторы** | | | | | | | | |
| **Мод. доступа** | | | | **Аргументы** | | **Назначение** | | |
| public | | | | Event[] events, Context context | | Создаёт объект. | | |
| **Методы** | | | | | | | | |
| **Имя** | **Мод. доступа** | | **Тип** | | **Аргументы** | | **Назначение** | |
| getCount | public | | int | | - | | Возвращает количество элементов в списке. | |
| getItem | public | | Event | | int position | | Возвращает элемент на заданной позиции. | |
| getItemId | public | | long | | int position | | Возвращает идентификатор элемента на заданной позиции. | |
| getView | public | | View | | int position, View convertView, ViewGroup parent | | Создаёт элементы пользовательского интерфейса. | |
| **Поля** | | | | | | | | |
| **Имя** | | **Мод. доступа** | | | **Тип** | | | **Назначение** |
| events | | private | | | Event[] | | | Массив событий. |
| inflater | | private | | | LayoutInflater | | | Объект, который позволяет создавать элементы графики. |

*HTMLScreenFragment*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | **Аргументы** | | | **Назначение** | |
| onCreateView | | public | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | | | Создаёт элементы пользовательского интерфейса. | |
| setText | | private | void | String text | | | Показывает текст. | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | **Тип** | **Назначение** | |
| viewForHTML | private | | | | TextView | Хранит view с текстом. | |

*InternalWebFragment*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| onCreateView | public | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | Создаёт элементы пользовательского интерфейса. |

*MapFragment*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | **Аргументы** | | | **Назначение** | |
| onCreate | | public | void | Bundle savedInstanceState | | | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. | |
| readStateFromBundle | | private | void | Bundle args | | | Считывает аргумент из Bumdle объекта, инициализирует поля класса | |
| onCreateView | | public | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | | | Создаёт элементы пользовательского интерфейса. | |
| handleMarkerClick | | private | void | Marker marker | | | Обрабатывает нажатие на отметку на карте. | |
| onSaveInstanceState | | public | void | Bundle outState | | | Сохраняет состояние фрагмента | |
| setArguments | | public | void | Bundle args | | | Задаёт аргументы для отображения фрагмента | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | **Тип** | **Назначение** | |
| MARKERS\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи массива отметок на карте. | |
| TITLE\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи заголовка фрагмента. | |
| markerWrappers | private | | | | MarkerWrapper[] | Массив отметок на карте. | |
| title | private | | | | String | Заголовок фрагмента. | |
| markersToWrappers | private | | | | Map<Marker, MarkerWrapper> | Map,определяющий MarkerWrapper по Markerю | |

*RSSFragment*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| showListOfRSSItems | private | View | LayoutInflater inflater,ViewGroup container | Отображает список RSS элементов. |
| openBrowser | private | void | String url | Открывает ссылку в браузере. |
| showEntireRSS | private | View | LayoutInflater inflater, ViewGroup container | Показывает RSS элемент с тексом. |
| onCreateView | public | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | Создаёт элементы пользовательского интерфейса. |

*RSSListAdapter*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Конструкторы** | | | | | | | | |
| **Мод. доступа** | | | | **Аргументы** | | **Назначение** | | |
| public | | | | Context context, RSSScreen[] rssItems | | Создаёт объект. | | |
| **Методы** | | | | | | | | |
| **Имя** | **Мод. доступа** | | **Тип** | | **Аргументы** | | **Назначение** | |
| getCount | public | | int | | - | | Возвращает количество элементов в списке. | |
| getItem | public | | RSSScreen | | int position | | Возвращает элемент на заданной позиции. | |
| getItemId | public | | long | | int position | | Возвращает идентификатор элемента на заданной позиции. | |
| getView | public | | View | | int position, View convertView, ViewGroup parent | | Создаёт элементы пользовательского интерфейса. | |
| **Поля** | | | | | | | | |
| **Имя** | | **Мод. доступа** | | | **Тип** | | | **Назначение** |
| rssItems | | private | | | RSSScreen[] | | | Массив событий. |
| inflater | | private | | | LayoutInflater | | | Объект, который позволяет создавать элементы графики. |

*ScreenFactory*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Конструкторы** | | | | | | | | |
| **Мод. доступа** | | | | **Аргументы** | | **Назначение** | | |
| private | | | | FragmentActivity activity, boolean isFirstFragment | | Создаёт объект. | | |
| **Методы** | | | | | | | | |
| **Имя** | **Мод. доступа** | | **Тип** | | **Аргументы** | | **Назначение** | |
| initFactory | public | | void | | FragmentActivity activity, boolean isFirstFragment | | Создаёт объект. | |
| instance | public | | ScreenFactory | | - | | Возвращает объект класса. | |
| changeFragments | public | | void | | FragmentManager manager, Fragment fragmentToAppear | | Заменяет текущий фрагмент на новый с анимацией. | |
| showFragment | public | | void | | BaseScreen view | | Создаёт и показывает новый фрагмент, если это требуется. | |
| openBrowser | private | | void | | String url | | Открывает ссылку в браузере. | |
| setFragment | private | | void | | Fragment fragmentToAppear | | Показывает новый фрагмент без анимации. | |
| openFile | private | | void | | ScreenWithFile fileView | | Открывает файл в стороннем приложении. | |
| getVkAccessToken | private | | AccessToken | | - | | Считыавет токен для ВКонтакте из SharedPreferences. | |
| getFacebookAccessToken | private | | AccessToken | | - | | Считыавет токен для Facebook из SharedPreferences. | |
| requestFacebookToken | private | | void | | - | | Запрашивает токен Facebook. | |
| requestVkToken | private | | void | | - | | Запрашивает токен ВКонтакте. | |
| notifyAboutAccessTokenReceiving | private | | void | | - | | Проверяет успешность получения токена. | |
| **Поля** | | | | | | | | |
| **Имя** | | **Мод. доступа** | | | **Тип** | | | **Назначение** |
| SHARED\_PREFERENCES\_TAG\_SOCIAL | | private | | | String | | | Тэг для SharerPreferences. |
| VK\_ACCESS\_TOKEN\_TAG | | private | | | String | | | Тэг для токена ВКонтакте в SharerPreferences. |
| FACEBOOK\_ACCESS\_TOKEN\_TAG | | private | | | String | | | Тэг для токена Facebook в SharerPreferences. |
| screenFactory | | private | | | ScreenFactory | | | Единственный объект класса. |
| activity | | private | | | FragmentActivity | | | Ссылка на активити. |
| isFirstFragment | | private | | | boolean | | | Указывает первый ли это фрагмент. |
| authenticationFragment | | private | | | AuthenticationFragment | | | Ссылка на фрагмент авторизации. |

*SocialNetworkAbstractList*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | **Аргументы** | | | **Назначение** | |
| onCreate | | public | void | Bundle savedInstanceState | | | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. | |
| readStateFromBundle | | private | void | Bundle args | | | Считывает аргумент из Bumdle объекта, инициализирует поля класса | |
| onSaveInstanceState | | public | void | Bundle outState | | | Сохраняет состояние фрагмента | |
| setArguments | | public | void | Bundle args | | | Задаёт аргументы для отображения фрагмента | |
| getGroupId | | protected | String | - | | | Возвращает идентификатор группы. | |
| getGroupName | | protected | String | - | | | Возвращает название группы. | |
| getRequester | | protected | AbstractRequester | - | | | Возвращает объект, который выполняет запросы к социальной сети. | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | **Тип** | **Назначение** | |
| GROUP\_NAME\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи названия группы. | |
| GROUP\_ID\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи идентификатора группы. | |
| REQUESTER\_TAG | private | | | | String | Используется как ключ в Bundle объекте для передачи объекта, осуществляющего запросы к социальной сети.. | |

*SocialNetworkAbstractList*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | **Аргументы** | | | **Назначение** | |
| onCreate | | public | void | Bundle savedInstanceState | | | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. | |
| readStateFromBundle | | private | void | Bundle args | | | Считывает аргумент из Bumdle объекта, инициализирует поля класса | |
| onSaveInstanceState | | public | void | Bundle outState | | | Сохраняет состояние фрагмента | |
| setArguments | | public | void | Bundle args | | | Задаёт аргументы для отображения фрагмента | |
| onCreateView | | public | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | | | Создаёт элементы пользовательского интерфейса. | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | **Тип** | **Назначение** | |
| GROUP\_ID\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи идентификатора группы. | |
| REQUESTER\_TAG | private | | | | String | Используется как ключ в Bundle объекте для передачи объекта, осуществляющего запросы к социальной сети. | |
| TITLE\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи заголовка фрагмента. | |
| STATE\_LISTENER\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи объекта для обратного вызова. | |
| groupId | private | | | | String | Хранит идентификатор группы. | |
| titleInput | private | | | | EditText | Вводимый пользователем заголовок. | |
| EditText | private | | | | EditText | Вводимый пользователем текст. | |
| actionBarTitle | private | | | | String | Заголовок, показываемый в ActionBar | |
| requester | private | | | | AbstractRequester | Объект, осуществляющий запросы к социальной сети. | |
| stateListener | private | | | | StateListener | Объект для обратного вызова. | |

*TopicsFragment*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | **Аргументы** | | | **Назначение** | |
| onCreate | | public | void | Bundle savedInstanceState | | | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. | |
| onSaveInstanceState | | public | void | Bundle outState | | | Сохраняет состояние фрагмента | |
| setArguments | | public | void | Bundle args | | | Задаёт аргументы для отображения фрагмента | |
| onViewCreated | | public | void | View view, Bundle savedInstanceState | | | Создаёт элементы пользовательского интерфейса. | |
| onCreateOptionsMenu | | public | void | Menu menu, MenuInflater inflater | | | Устанавливает меню в ActionBar | |
| onOptionsItemSelected | | public | boolean | MenuItem item | | | Обрабатывает нажатия на меню в ActionBar. | |
| onResume | | public | void | - | | | Служит для создания восстановления объекта. | |
| setAdapter | | private | void | - | | | Устанавливает адаптер для списка. | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | **Тип** | **Назначение** | |
| VK\_TOPICS\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи объекта с темой. | |
| COMMENTS\_STATE | private | | | | String | Используется как ключ в Bundle объекте для передачи объекта для оповещения об изменении состояния. | |
| topics | private | | | | SocialNetworkTopic[] | Список обсуждений. | |
| adapter | private | | | | TopicsAdapter | Адаптер для списка обсуждений. | |
| stateChanged | private | | | | StateListener | Состояние списка. | |

*TopicsFragment*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | **Аргументы** | | | **Назначение** | |
| onCreate | | public | void | Bundle savedInstanceState | | | При повторном показе фрагмента извлекает данные из Bundle и восстанавливает состояние фрагмента. | |
| onSaveInstanceState | | public | void | Bundle outState | | | Сохраняет состояние фрагмента | |
| setArguments | | public | void | Bundle args | | | Задаёт аргументы для отображения фрагмента | |
| onCreateView | | public | void | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState | | | Создаёт элементы пользовательского интерфейса. | |
| onViewCreated | | public | void | View view, Bundle savedInstanceState | | | Создаёт элементы пользовательского интерфейса. | |
| loadComments | | private | vois | - | | | Запрашивает загрузку комментариев. | |
| instantiateAdapte | | private | vois | - | | | Устанавливает адаптер к списку. | |
| createHeaderView | | private | void | - | | | Создаёт элемент, который показывается вверху списка. | |
| createFooterView | | private | void | - | | | Создаёт элемент, который показывается внизу списка. | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | **Тип** | **Назначение** | |
| WALL\_COMMENTS\_TAG | public | | | | String | Используется как ключ в Bundle объекте для передачи списка комментариев. | |
| WALL\_COMMENTS\_POST\_TAG | private | | | | String | Используется как ключ в Bundle объекте для передачи объекта обсуждения. | |
| COMMENTS\_LISTENER\_TAG | private | | | | String | Используется как ключ в Bundle объекте для передачи объекта, которые следит за изменениями в комментариях. | |
| TYPED\_COMMENT | private | | | | String | Используется как ключ в Bundle объекте для сохранения напечатанного комментария. | |
| post | private | | | | SocialNetworkTopic | Пост. | |
| comments | private | | | | SocialNetworkEntry[] | Массив комментариев. | |
| stateListener | private | | | | AtomicBoolean | Состояние списка. | |
| containerWidth | private | | | | int | Ширина контейнера. | |
| commentText | private | | | | String | Комментарий, набранный пользователем. | |
| footerView | private | | | | View | Хранит элемент внизу списка комментариев. | |
| input | private | | | | EditText | Поле для ввода комментария. | |
| headerView | private | | | | View | Хранит элемент вверху списка комментариев. | |
| stateListener | private | | | | StateListener | Состояние списка. | |

*BaseScreen*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Конструкторы** | | | | | | | | | | |
| **Мод. доступа** | | | **Аргументы** | | | | **Назначение** | | | |
| protected | | | **-** | | | | **-** | | | |
| protected | | | JSONObject jsonObject | | | | Считывает поля, которые всегда присутствуют в JSON объекте | | | |
| protected | | | JSONObject jsonObject, String serverURL | | | | Служит для создания объектов, которым важно знать адрес сервера. | | | |
| protected | | | Parcel input | | | | Служит для восстановления сериализованного объекта | | | |
| **Методы** | | | | | | | | | | |
| **Имя** | | **Мод. доступа** | | **Тип** | **Аргументы** | | | | **Назначение** | |
| getScreen | | public | | BaseScreen | String json, String serverURL | | | | Парсит входной JSON и создаёт граф разделов. | |
| getUrl | | public | | String | - | | | | Возвращает ссылку. | |
| getDescription | | public | | String | - | | | | Возвращает описание раздела. | |
| getScreenType | | public | | int | - | | | | Возвращает тип раздела. | |
| getName | | public | | String | - | | | | Возвращает название раздела. | |
| getKey | | public | | String | - | | | | Возвращает уникальный идентификатор раздела. | |
| parseUsualDescriptionOfTheView | | private | | void | JSONObject jsonObject | | | | Считывает поля из JSON. | |
| parseJSON | | private | | void | JSONArray jsonArray, String serverURL | | | | Перебирает все данные JSON объекты и создаёт разделы. | |
| getScreenElements | | public | | BaseScreen[] | - | | | | Возвращает подразделы. | |
| isMainScreen | | public | | boolean | - | | | | Проверяет, является ли данный раздел главным. | |
| getDescriptionsOfFiles | | public | | void | Collection<FileDescription> descriptions | | | | Рекурсивно перебирает подразделы, заполняя список файлов. | |
| describeContents | | public | | int | - | | | | Используется при сериализации. | |
| writeToParcel | | public | | void | Parcel dest, int flags | | | | Используется при сериализации. | |
| **Поля** | | | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | | **Тип** | | **Назначение** | |
| CREATOR | pubcic | | | | | Creator<BaseScreen> | | Используется при десериализациии. | |
| url | protected | | | | | String | | Ссылка на страницу, описывающую раздел. | |
| screenType | protected | | | | | int | | Тип раздела. | |
| isMainView | private | | | | | boolean | | Является ли раздел главным. | |
| name | private | | | | | String | | Название раздела. | |
| key | private | | | | | String | | Уникальный идентификатор раздела. | |
| childViews | private | | | | | BaseScreen[] | | Массив подразделов | |
| description | private | | | | | String | | Описание раздела. | |

*Event*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Конструкторы** | | | | | | | | | | |
| **Мод. доступа** | | | **Аргументы** | | | | **Назначение** | | | |
| protected | | | JSONObject jsonObject | | | | Считывает необходимые поля из JSON объекта. | | | |
| protected | | | Parcel input | | | | Служит для восстановления сериализованного объекта | | | |
| **Методы** | | | | | | | | | | |
| **Имя** | | **Мод. доступа** | | **Тип** | **Аргументы** | | | | **Назначение** | |
| describeContents | | public | | int | - | | | | Используется при сериализации. | |
| writeToParcel | | public | | void | Parcel dest, int flags | | | | Используется при сериализации. | |
| getTelephone | | public | | String | - | | | | Возвращает номер телефона. | |
| getAddress | | public | | String | - | | | | Возвращает адрес события. | |
| getDate | | public | | Date | - | | | | Возвращает дату события. | |
| **Поля** | | | | | | | | | | |
| **Имя** | **Мод. доступа** | | | | | **Тип** | | **Назначение** | |
| CREATOR | pubcic | | | | | Creator<BaseScreen> | | Используется при десериализациии. | |
| telephone | private | | | | | String | | Номер телефона. | |
| address | private | | | | | String | | Адресс события. | |
| date | private | | | | | Date | | Дата события. | |

*EventScreen*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Конструкторы** | | | | | | |
| **Мод. доступа** | | **Аргументы** | | | **Назначение** | |
| public | | JSONObject jsonObject, String serverURL | | | Считывает необхожимые поля из JSON объекте | |
| protected | | Parcel input | | | Служит для восстановления сериализованного объекта | |
| **Методы** | | | | | | |
| **Имя** | **Мод. доступа** | | **Тип** | **Аргументы** | | **Назначение** |
| describeContents | public | | int | - | | Используется при сериализации. |
| writeToParcel | public | | void | Parcel dest, int flags | | Используется при сериализации. |
| getFileDescription | public | | FileDescription | - | | Возвращает описание файла. |
| getEvents | public | | Event[] | - | | Возвращает список событий. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Изм. | Номера листов | | | | Всего листов (страниц) в докум. | № докум. | Входящий №  сопроводительного докум. и дата | Подп. | Дата |
| изме-ненных | заме-ненных | новых | анну-лиро-ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |