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. |

## 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. |

# ПРИЛОЖЕНИЕ 2

# ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ

*ConformanceCheckingTransformation*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| transformForConformanceChecking | public | Object[] | UIPluginContext context, Petrinet petrinet, XLog log | Работает с входными и выходными данными, связывает графический интерфейс программы. |
| performTransformation | private | Object[] | XFactory factory, Map<Transition, Set<Activity>> justTransitions | Выполняет преобразование модели и лога исходя из входных данных. |
| showPetriNet | private | TaskListener.InteractionResult | - | Предоставляет возможность продемонстрировать пользователю исходную сеть Петри. |
| showDefinePartition | private | TaskListener.InteractionResult | Map<Transition, Set<Activity>> justTransitions,  Set<Activity> lowLevelTransitionNames | Отвечает за показ окна с возможностью выбора разбиения. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Назначение** |
| events | private | List<ArrayList<Activity>> | Хранит лог событий |
| context | private | UIPluginContext | Необходимо для воздействия плагина и Фреймворка. |
| petrinet | private | PetriNet | Хранит сеть Петри с которой осуществляется работа. |

*ConverterForLowLevelTrace*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| toHighLevel | public | List<Transition> | List<Activity> lowLevelTrace, Map<Transition, Set<Activity>> highLevelActivities | Преобразует последовательность низкоуровневых действий к последовательности высокоуровневых. |
| search | private | Transition | Map<Transition, Set<Activity>> highLevelActivities, Activity lowLevelActivity | Ищет среди высокоуровневых действий соответствующее данному низкоуровневому, используя разбиение |

*Activity*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | | |
| **Имя** | | **Мод. доступа** | **Тип** | | **Аргументы** | | **Назначение** | |
| compareTo | | public | int | | Activity o | | Сравнивает два объекта типа Activity по их имени. | |
| equals | | public | boolean | | Object o | | Сравнивает два объекта типа Activity: возвращает true, если они равны и false в обратном случае. | |
| setTime | | public | void | | String time | | Устанавливает время, в которое закончилось действие. | |
| setLifecycle | | public | void | | String lifecycle | | Устанавливает жизненный цикл действия. | |
| **Поля** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | **Тип** | | **Назначение** | | |
| name | private | | | String | | Хранит имя действия. | | |
| lifecycle | private | | | String | | Хранит жизненный цикл действия. | | |
| time | private | | | String | | Хранит время выполнения действия. | | |
| **Свойства** | | | | | | | | |
| **Имя** | **Мод. доступа** | | | **Тип** | | | | **Назначение** |
| hashCode | public | | | int | | | | Возвращает хэш код действия. |
| getTime | public | | | String | | | | Возвращает время выполнения действия. |
| getLifecycle | public | | | String | | | | Возвращает жизненный цикл действия. |

*PartitionsPanel*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | | |
| **Имя** | **Мод. доступа** | | **Тип** | | **Аргументы** | **Назначение** | |
| fillColumnsWithEmptyLists | private | | void | | - | Заполняет столбец пустыми значениями. | |
| fillResourcesColumn | private | | void | | Set<Activity> lowLevelTransitionNames | Заполняет столбец с ресурсами для выбора разбиений. | |
| **Поля** | | | | | | | |
| **Имя** | | **Мод. доступа** | | **Тип** | | | **Назначение** |
| highColumn | | private | | OneChoiceSelectPanel<Transition> | | | Хранит набор высокоуровневых переходов, отображаемых в панели выбора разбиений. |
| resourceColumn | | private | | MultipleSelectPanel<Activity> | | | Хранит набор действий, отображаемых в панели выбора разбиений |
| mapping | | private | | Map<Transition, Set<Activity>> | | | Хранит разбиение (соответствие между высокоуровневыми переходами и множествами низкоуровневых действий). |
| contentPanel | | private | | JPanel | | | Хранит визуальную панель с размещенными компонентами. |
| allNonSelectedResources | | private | | Collection<Activity> | | | Хранит множество всех невыделенных низкоуровневых действий |
| excludedActivities | | private | | Set<Activity> | | | Хранит множество действий, которые необходимо показывать для выделенного перехода. |

*ConverterForModel*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| addLoop | private | void | Petrinet petrinet, Transition transition, AtomicInteger h | Добавляет петлю к выбранному переходу в заданную сеть. |
| makeNewModel | private | void | Petrinet petrinet, List<Transition> log | Производит преобразование модели. |
| howManyTransitionWithThisName | private | int | Petrinet petrinet, Transition transition | Возвращает количество переходов в сети с указанным именем. |

*TransitionsColumnSelectionListener*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| selected | private | void | Transition selectedOption | Обрабатывает событие выделения высокоуровневого действия и заполняет форму в соответствии с ним. |
| deselected | private | void | Transition deselectedOption | Обрабатывает событие снятия выделения высокоуровневого действия и заполняет форму в соответствии с ним. |
| selectionChanged | private | void | Transition previousSelectedTransition, Transition newSelectedTransition | Выполняет связку действий выделения и снятия выделения. |

*ResourcesColumnListener*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| **Имя** | **Мод. доступа** | **Тип** | **Аргументы** | **Назначение** |
| selected | private | void | Activity selectedResource | Обрабатывает событие выделения низкоуровневого действия и заполняет форму в соответствии с ним. |
| deselected | private | void | Activity deselectedResource | Обрабатывает событие снятия выделения низкоуровневого действия и заполняет форму в соответствии с ним. |
| selectionChanged | private | void | Activity previousSelectedResource, Activity newSelectedResource | Выполняет связку действий выделения и снятия выделения. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |