

Shared Preferences

android



Root-права

Для 100% погружения в новую тему рекомендуется выставить на устройстве админские права!

Root-права (или супер-юзер) – Предоставляют владельцу Android-устройства полный контроль над системой, что позволяет редактировать и изменять системные файлы и папки.

Программы для root-доступа

Root-доступ можно получить при помощи специальных программ. Если антивирус будет показывать уведомление о том, что программа заражена, не нужно обращать на это внимание. Такие программы требуют доступ к ядру устройства, это по сути является взломом, и именно по этой причине антивирусы выдают предупреждение.

Требуются root-права...

Практически каждый, кто купил смартфон или планшет на OS Android, первым делом начинает искать различные программы и игры на него. И часто, при просмотре программ, встречаются такие фразы, как "Требуются root права, Нужен root, Для работы программы необходимо получить root.." и тд.

Так для чего же нужны root-права? Как их получить? Что они дают?

Что дают root-права?

- Изменять системные файлы
- Удалять стандартные приложения (календарь, карты и тд.)
- Изменять и удалять темы, ярлыки
- Устанавливать изображения, выводимые на дисплей устройства при его загрузке
- Добавлять исполняемые файлы Linux
- Оптимизировать работу своего устройства, разогнать процессор
- Запускать приложения, которые не работают без активации root-прав
- Осуществлять резервное копирование данных
- Устанавливать модифицированные прошивки
- И множество других возможностей...

Гарантийное обслуживание

С установкой root-прав на смартфоне или планшете, теряется право на гарантийное обслуживание. Если впоследствии с аппаратом возникнут какие-либо проблемы, в сервисном центре вполне обоснованно откажут в гарантийном обслуживании. Кроме того, если совершить критические ошибки во время процедуры рутования, можно превратить свой любимый смартфон в бесполезный кусок пластмассы, выведя из строя программную часть.

祝你健康快樂好心情的管理員權

- <https://www.youtube.com/watch?v=4zygUn1mkuw>
- https://www.youtube.com/watch?v=uKE_UDYmBvo

Увидев это, я чёт передумал ставить рут 😊



成功连接电脑

现在可以用刷机大师来管理您的手机
也可以继续在应用酷



装机必备

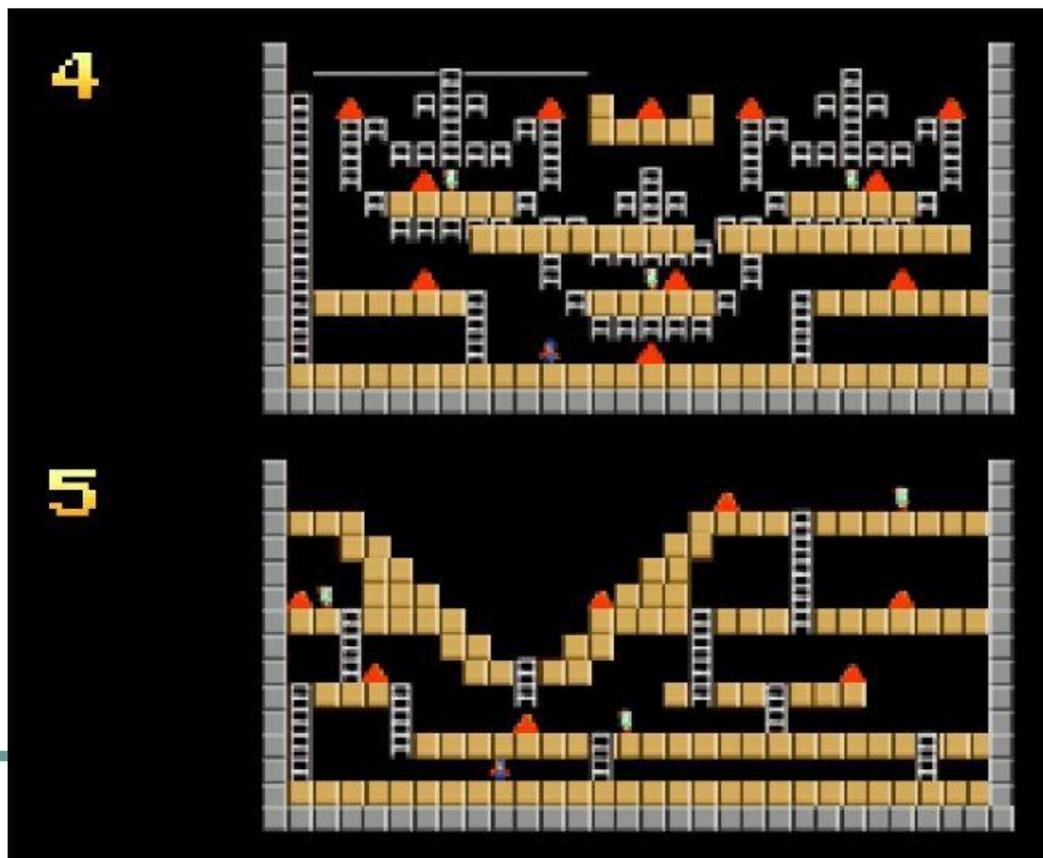
趣味发现

专题汇聚

每日推荐

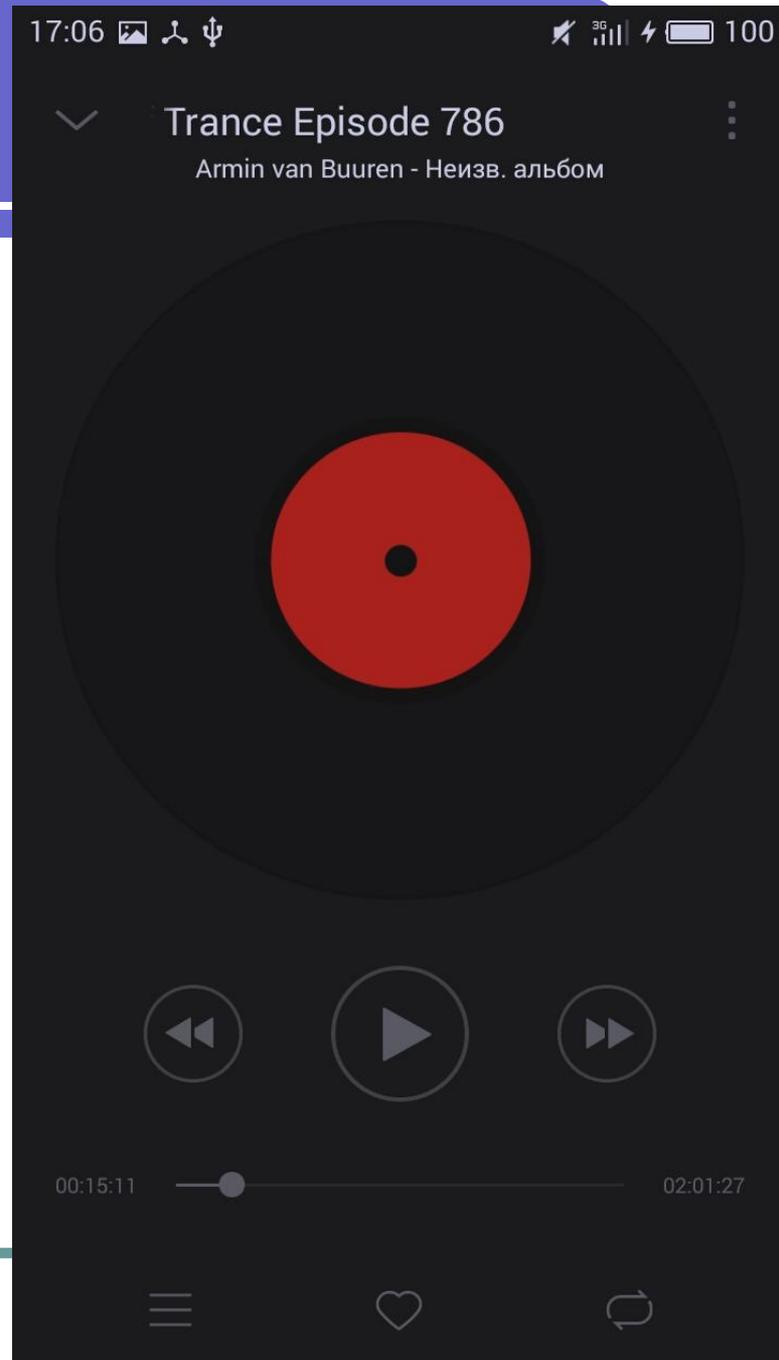
Сохранение данных

Очень часто требуется сохранить какие-нибудь мелкие настройки, вроде того – до какого уровня дошёл пользователь в игре Load Runner...



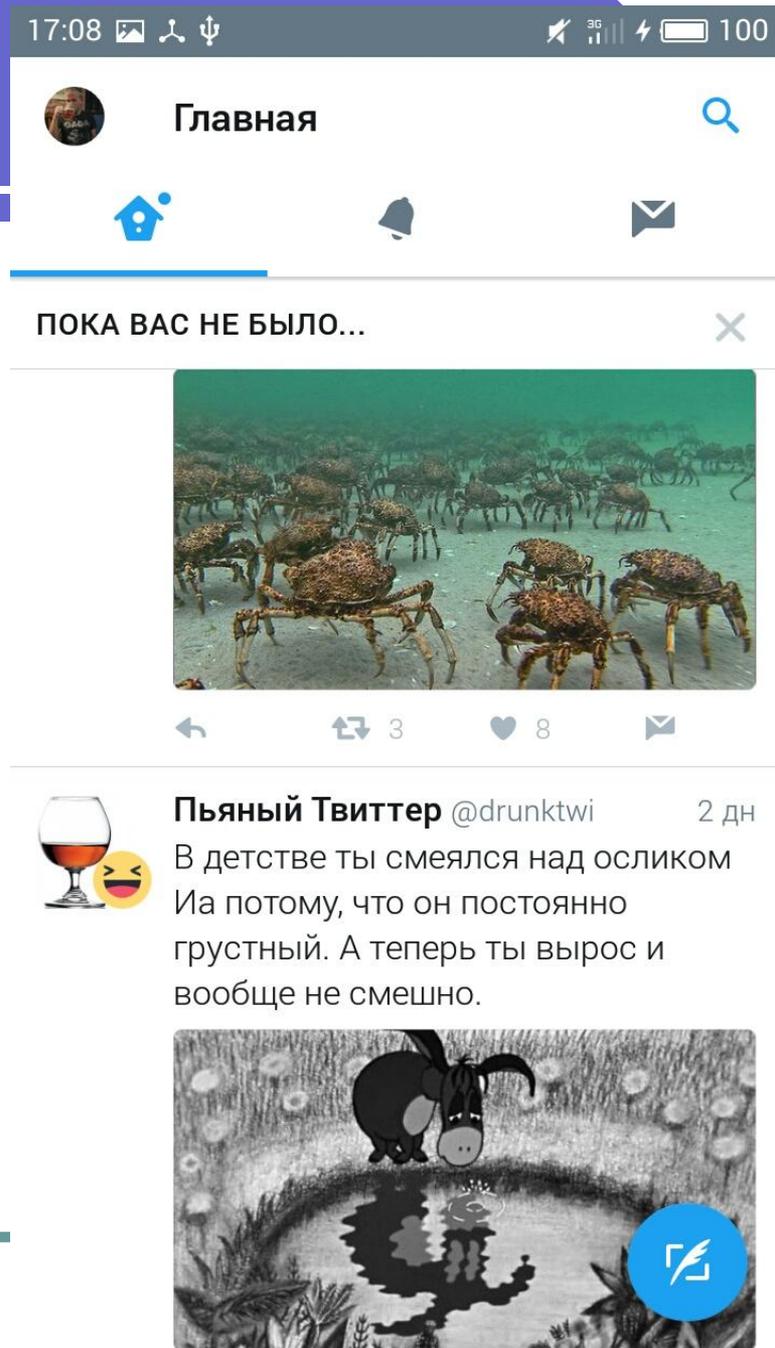
Сохранение данных

На каком треке, и какой секунде воспроизведения остановился плеер...



Сохранение данных

Какой именно твит пользователь читал в прошлый раз...



17:08 [icons] 100

Главная [search icon]

ПОКА ВАС НЕ БЫЛО... [close icon]



← ↻ 3 ❤️ 8 ✉

 **Пьяный Твиттер** @drunktwi 2 дн

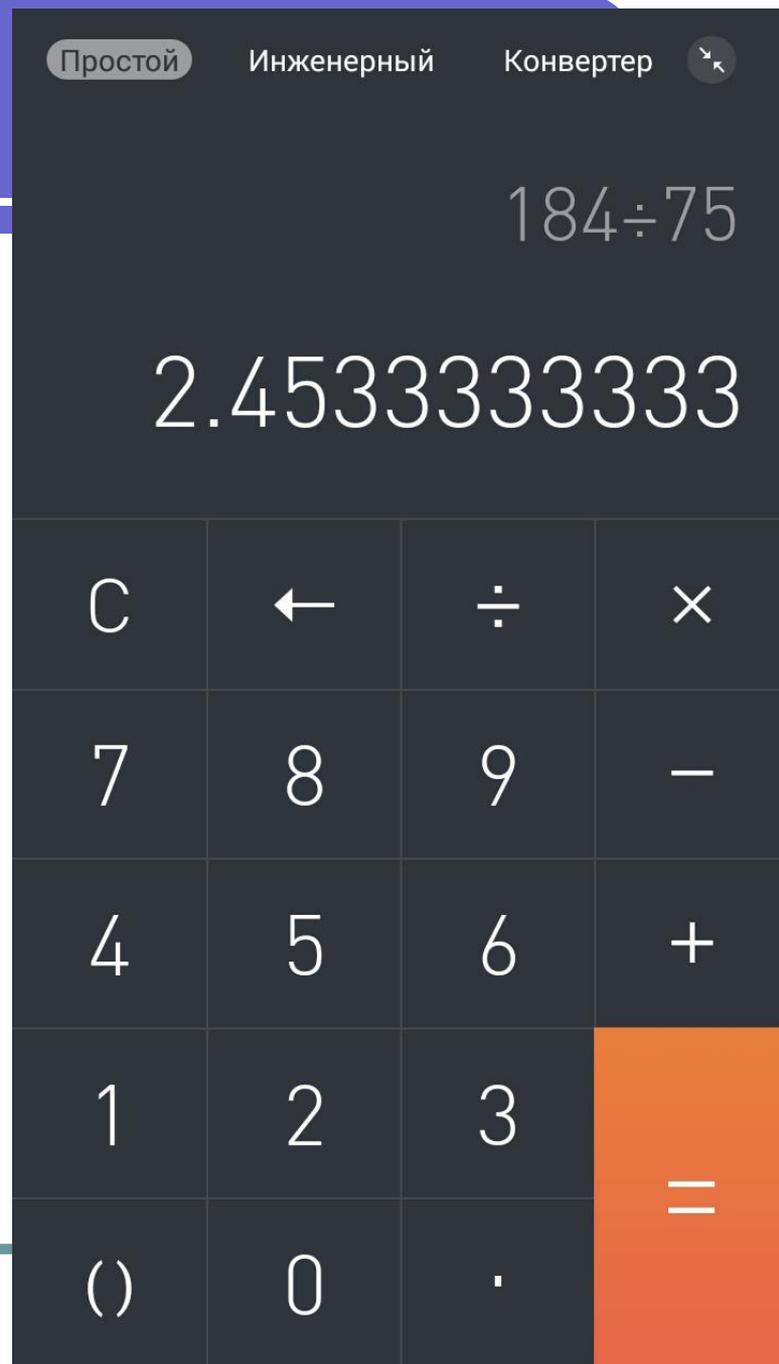
В детстве ты смеялся над осликом Иа потому, что он постоянно грустный. А теперь ты вырос и вообще не смешно.



[share icon]

Сохранение данных

И даже калькулятор следит за пользователем и тихонечко запоминает, что на нём считали в последнее время...



Как сохранить данные

Система Android предлагает три способа сохранения информации:

- Работа с файлами в стиле Java
- Создание базы данных
- **Shared Preferences**

Плюсы и минусы способов

Хранить данные в базе не всегда оправдано, особенно если данных не слишком много, они простые и не нужно анализировать данные на предмет, сколько уровней игры прошёл пользователь в прошлом году, и сколько часов листал ленту твиттера за октябрь и ноябрь.

Вариант с файлами хорош, например, для сохранения больших текстовых заметок. Естественно, и для простых данных тоже можно использовать файлы. Записали что-то в файл, а потом открыли его и считали данные.

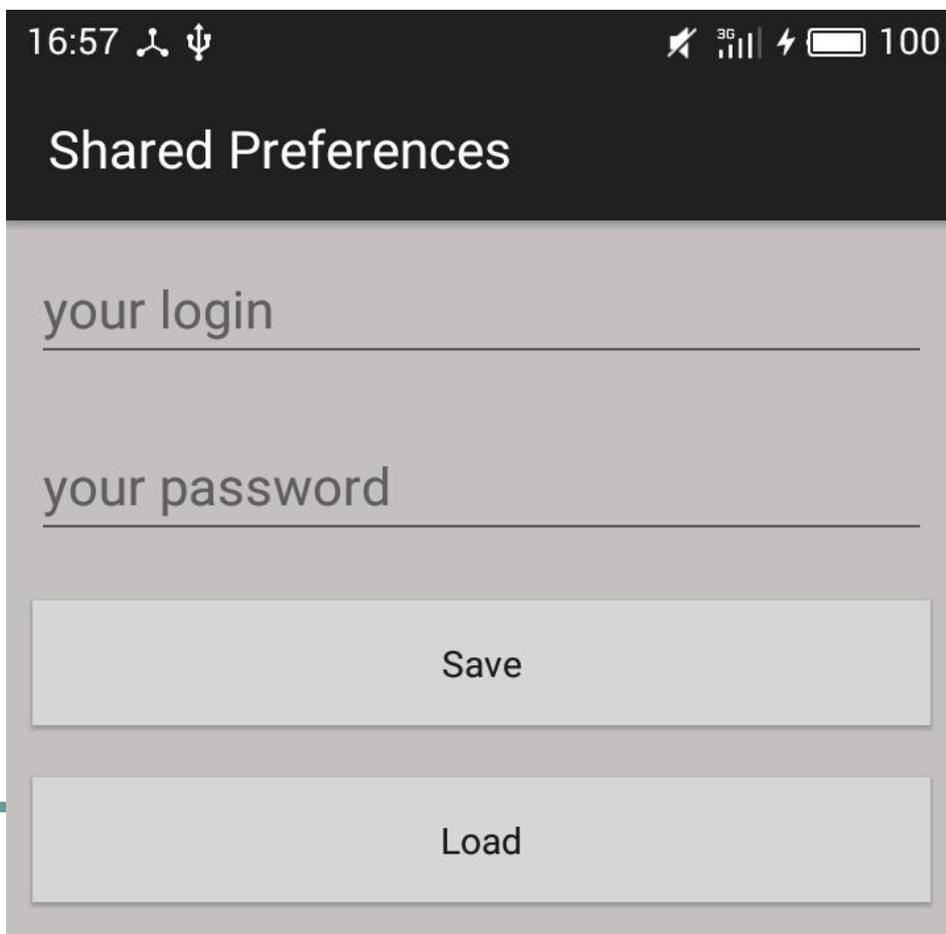
Но на самом деле не нужно изобретать велосипед и придумывать свою структуру для хранения данных, в Android существует класс **SharedPreferences**, разработанный специально для этих целей. Приложение автоматически создаёт файл в своей папке и хранит простые **данные в виде «ключ — значение»**. Весь процесс создания, открытия, чтения файла оптимизирован.

Shared Preferences

Общие настройки поддерживают базовые типы **boolean**, **String**, **float**, **long** и **int**, что делает их идеальным средством для быстрого сохранения значений по умолчанию, полей объектов класса, текущего состояния графического интерфейса и пользовательских настроек.

Пример кода

<https://git.io/vXJ5t>



16:57 人 ψ 3G 100

Shared Preferences

your login

your password

Save

Load

The image shows a mobile application interface for 'Shared Preferences'. At the top, there is a status bar with the time '16:57', a person icon, a psi symbol, '3G' signal strength, and a battery icon at '100'. Below the status bar is a dark header with the text 'Shared Preferences'. The main content area has a light gray background and contains two text input fields. The first field is labeled 'your login' and the second is labeled 'your password'. Below these fields are two large, light gray buttons: 'Save' and 'Load'.

Получение общих настроек

getPreferences

```
SharedPreferences getPreferences (int mode)
```

Retrieve a `SharedPreferences` object for accessing preferences that are private to this activity. This simply calls the underlying `getSharedPreferences(String, int)` method by passing in this activity's class name as the preferences name.

Parameters

mode	int: Operating mode. Use <code>MODE_PRIVATE</code> for the default operation.
------	---

Чтобы внести изменения в настройки, нужно использовать класс **SharedPreferences.Editor**. Получить объект **Editor** можно через вызов метода **edit()** объекта **SharedPreferences**. После того, как внесены все необходимые изменения, вызывается метод **apply()**, чтобы изменения вступили в силу.

Способы получения

Чтобы получить экземпляр класса **SharedPreferences** для получения доступа к настройкам в коде приложения используется один из трёх методов:

- **getPreferences()** — внутри активити, чтобы обратиться к определённому предпочтению в рамках одного активити
- **getSharedPreferences()** — внутри активити, чтобы обратиться к предпочтению на уровне приложения
- **getDefaultSharedPreferences()** — из объекта `PreferencesManager`, чтобы получить общедоступную настройку, предоставляемую Android

Как сохранить настройки

Public Methods

abstract void	<code>apply()</code> Commit your preferences changes back from this Editor to the <code>SharedPreferences</code> object it is editing.
abstract <code>SharedPreferences.Editor</code>	<code>clear()</code> Mark in the editor to remove <i>all</i> values from the preferences.
abstract boolean	<code>commit()</code> Commit your preferences changes back from this Editor to the <code>SharedPreferences</code> object it is editing.
abstract <code>SharedPreferences.Editor</code>	<code>putBoolean(String key, boolean value)</code> Set a boolean value in the preferences editor, to be written back once <code>commit()</code> or <code>apply()</code> are called.
abstract <code>SharedPreferences.Editor</code>	<code>putFloat(String key, float value)</code> Set a float value in the preferences editor, to be written back once <code>commit()</code> or <code>apply()</code> are called.
abstract <code>SharedPreferences.Editor</code>	<code>putInt(String key, int value)</code> Set an int value in the preferences editor, to be written back once <code>commit()</code> or <code>apply()</code> are called.
abstract <code>SharedPreferences.Editor</code>	<code>putLong(String key, long value)</code> Set a long value in the preferences editor, to be written back once <code>commit()</code> or <code>apply()</code> are called.
abstract <code>SharedPreferences.Editor</code>	<code>putString(String key, String value)</code> Set a String value in the preferences editor, to be written back once <code>commit()</code> or <code>apply()</code> are called.
abstract <code>SharedPreferences.Editor</code>	<code>putStringSet(String key, Set<String> values)</code> Set a set of String values in the preferences editor, to be written back once <code>commit()</code> or <code>apply()</code> is called.
abstract <code>SharedPreferences.Editor</code>	<code>remove(String key)</code> Mark in the editor that a preference value should be removed, which will be done in the actual preferences once <code>commit()</code> is called.

Получение настроек

Все эти методы возвращают экземпляр класса **SharedPreferences**, из которого можно получить соответствующую настройку с помощью ряда методов:

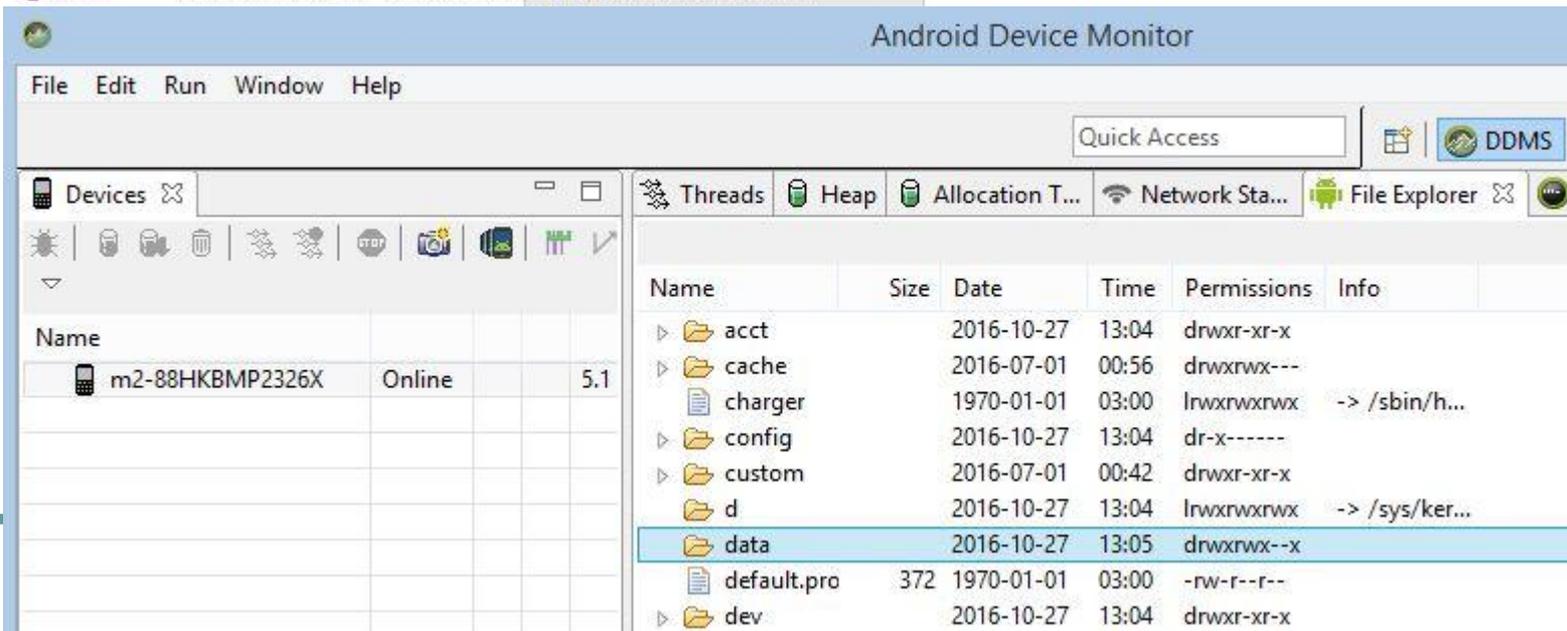
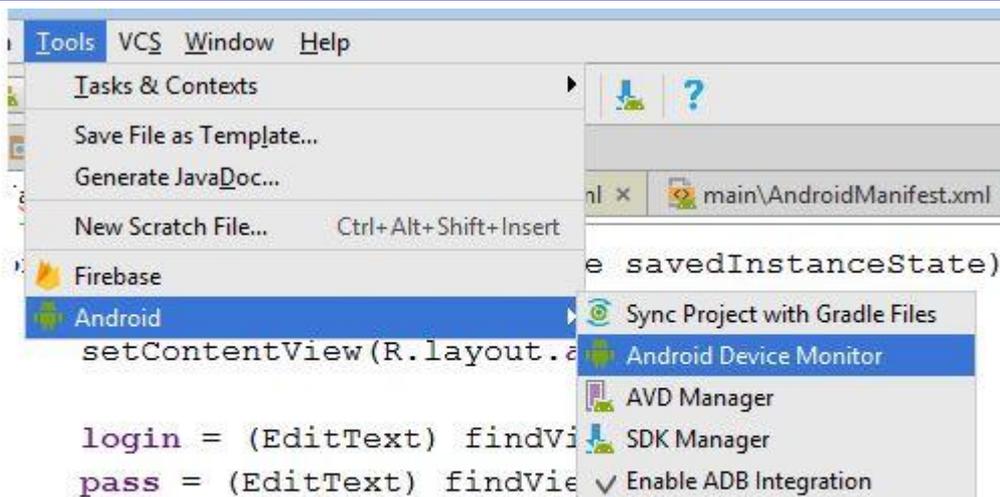
```
getAll ()                               Map<String, ?>
getBoolean (String s, boolean b)        boolean
getFloat (String s, float v)            float
getInt (String s, int i)                int
getLong (String s, long l)              long
getString (String s, String s1)         String
getStringSet (String s, Set<String> set) Set<String>
```

Где хранятся настройки

Файлы настроек хранятся в каталоге `/data/data/имя_пакета/shared_prefs/имя_файла_настроек.xml`.

Чтобы увидеть этот xml-файл, в котором хранятся все ключи и значения shared preferences, необходимо зайти в Android Device Monitor, выбрать там вкладку File Manager, найти папку data, и развернуть её. Во вложенной папке data будет файл shared preferences. Но, **для этого нужны root-права на устройстве!**

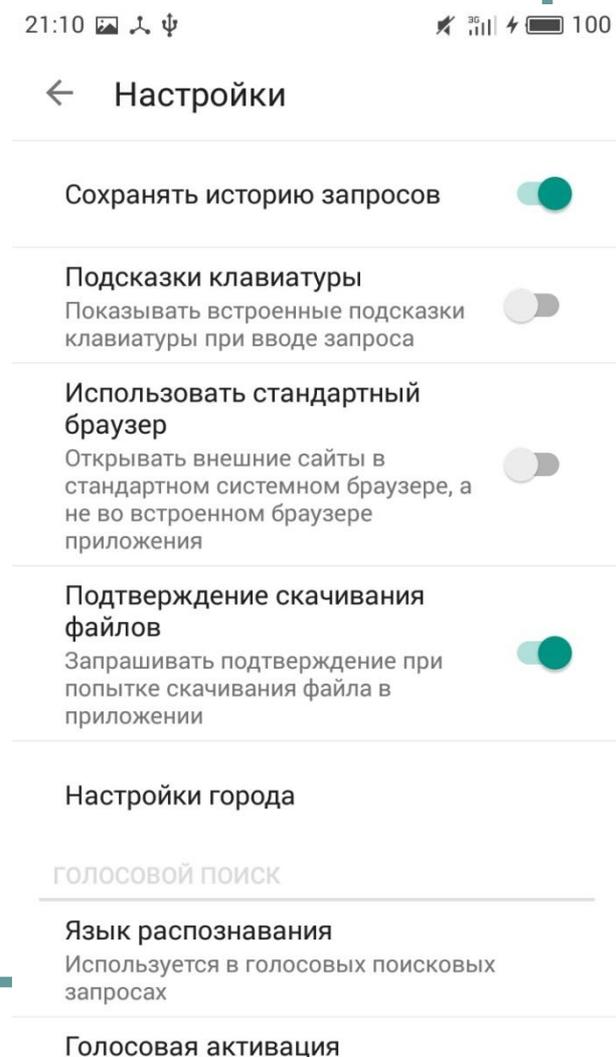
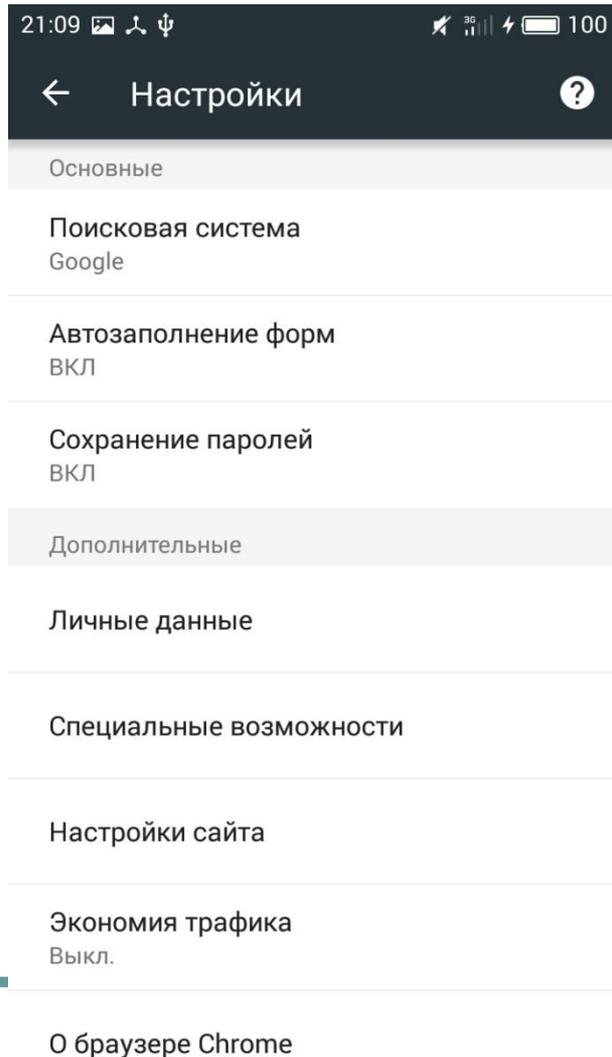
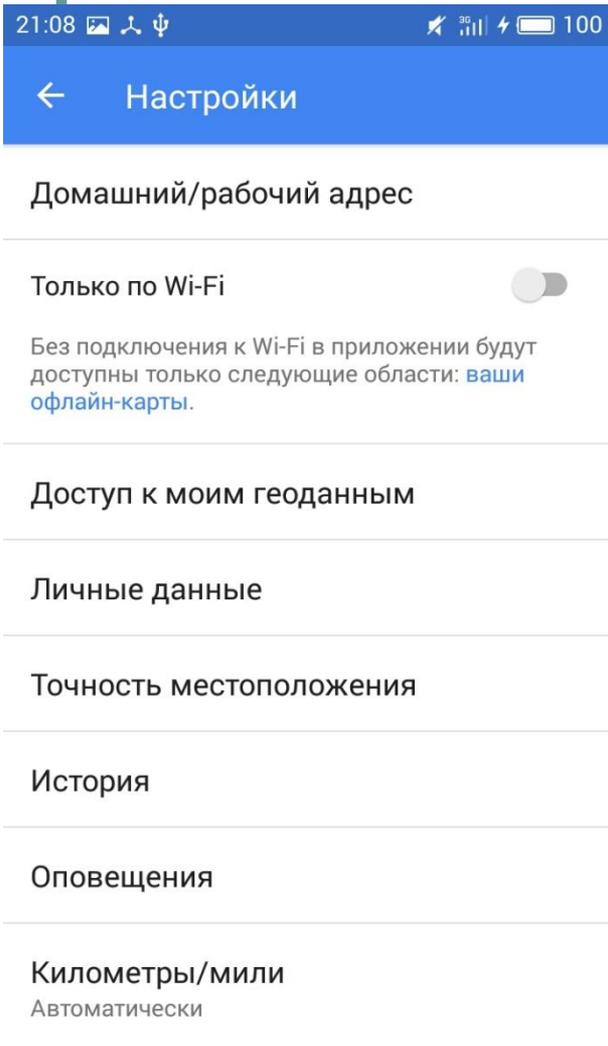
Где хранятся настройки



Preferences Framework

Иногда необходимо предоставить пользователю возможность настраивать функциональность приложения под свои вкусы. Часто для этих целей в приложениях используют отдельный экран **настроек**. Безусловно, можно создать собственный xml-файл разметки и разместить там нужные элементы управления для активности с настройками, но Android для этих целей предоставляет собственный **Preferences Framework**, с помощью которого можно создать индивидуальный набор предпочтений и встраивать их в приложения. Главное преимущество использования фреймворка состоит в том, что экран настроек выглядит стандартно во всех приложениях и позволяет пользователю быстрее разобраться в работе.

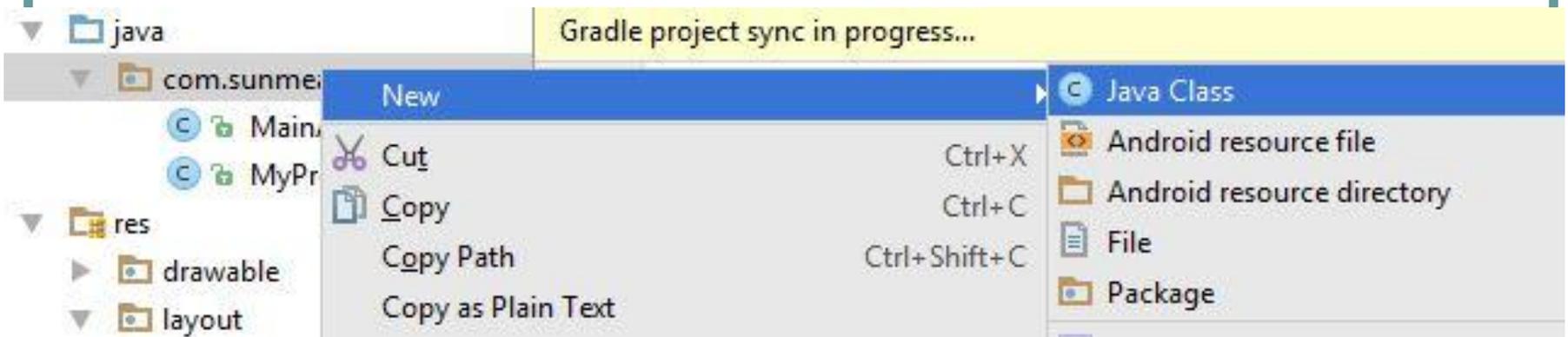
Как выглядят экраны настроек



PreferenceActivity

Использование фреймворка позволяет быстро создать экран настроек, практически без написания кода. Предпочтения — это отдельная активность в приложении. Сама активность настроек наследуется от класса **PreferenceActivity**, а не от стандартного класса **Activity**. Предпочтения определяются в отдельном XML-файле, где корневым элементом является элемент `<PreferenceScreen>`, который представляет собой контейнер для предпочтений и может содержать дочерние элементы `<PreferenceScreen>`. Также существует контейнерный элемент `<PreferenceCategory>`, который предназначен для объединения предпочтений в группы.

Создание активности настроек



Create New Class

Name:

Kind:

Superclass:

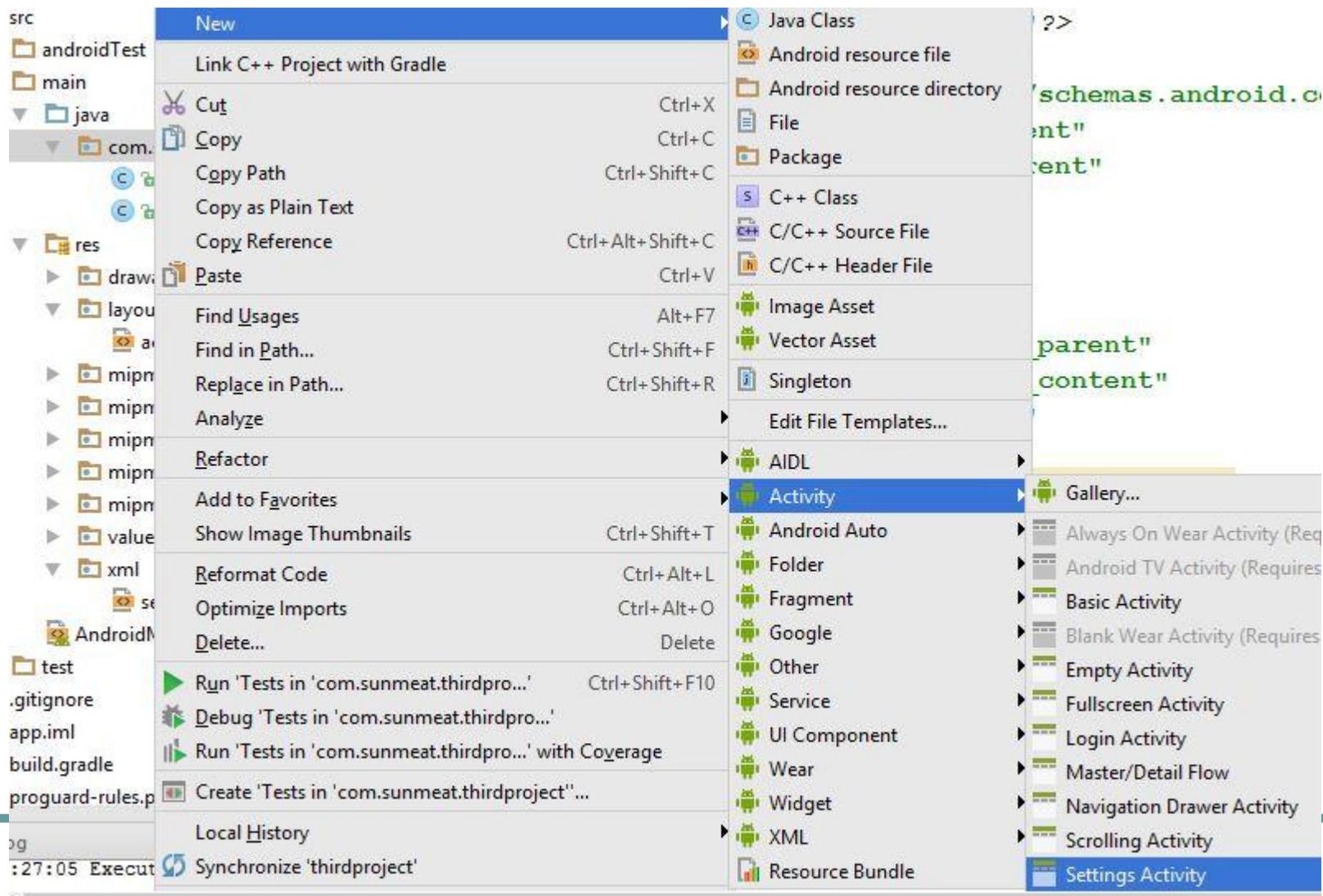
Interface(s):

Package:

Visibility: Public Package Private

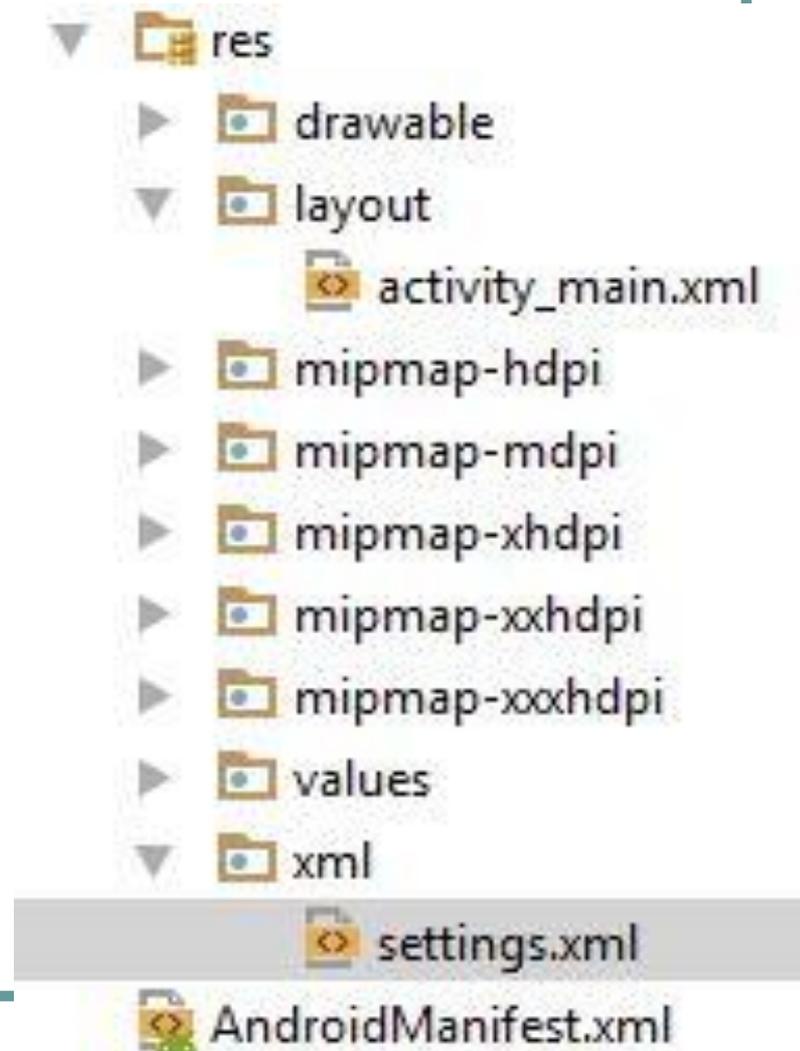
Modifiers: None Abstract Final

Этот шаблон слишком сложный!



Разметка пунктов меню

Определение пунктов меню настроек должно находиться в XML-файле с любым названием, который нужно положить в папку `res / xml` (папку `xml` придётся создать).



Пример кода

<https://git.io/vXU6L>

22:16   

 3G  100

Preferences Framework

Выставить настройки

Прочитать настройки

22:16  

 3G  100

первая категория настроек

первый пункт настроек

описание пункта с чекбоксом



второй пункт настроек

это пункт с текстовым полем, нажмите и проверьте!

вторая категория настроек

третий пункт настроек

это пункт со списком вариантов, нажмите и проверьте!

рычажок включён



пятый пункт настроек

перейти к дополнительным настройкам

крутой шестой пункт

влияет на системные настройки

седьмой пункт

влияет на восьмой пункт!



восьмой пункт

работает, если есть галочка на 7 пункте