**Настройки приложения (Preferences)**

**Создание и получение настроек**

Нередко приложению требуется сохранять небольшие кусочки данных для дальнейшего использования, например, данные о пользователе, настройки конфигурации и т.д. Для этого в Android существует концепция Preferences или настройки. Настройки представляют собой группу пар ключ-значение, которые используются приложением.

В качестве значений могут выступать данные следующих типов: **Boolean, Float, Integer, Long, String,** набор строк.

Настройки общими для всех activity в приложении, но также могут быть и настройки непосредственно для отдельных activity

**Общие настройки**

Для работы с разделяемыми настройками в классе Activity (точнее в его базовом классе Context) имеется метод **getSharedPreferences()**:

import android.content.SharedPreferences;

//...........................

SharedPreferences settings = getSharedPreferences("PreferencesName",

MODE\_PRIVATE);

Первый параметр метода указывает на название настроек. В данном случае название - "PreferencesName". Если настроек с подобным названием нет, то они создаются при вызове данного метода. Второй параметр указывает на режим доступа. В данном случае режим описан константой MODE\_PRIVATE

Класс android.content.SharedPreferences предоставляет ряд методов для управления настройками:

* contains(String key): возвращает true, если в настройках сохранено значение с ключом key
* getAll(): возвращает все сохраненные в настройках значения
* getBoolean (String key, boolean defValue): возвращает из настроек значение типа Boolean, которое имеет ключ key. Если элемента с таким ключом не окажется, то возвращается значение defValue, передаваемое вторым параметром
* getFloat(String key, float defValue): возвращает значение типа float с ключом key. Если элемента с таким ключом не окажется, то возвращается значение defValue
* getInt(String key, int defValue): возвращает значение типа int с ключом key
* getLong(String key, long defValue): возвращает значение типа long с ключом key
* getString(String key, String defValue): возвращает строковое значение с ключом key
* getStringSet(String key, Set<String> defValues): возвращает массив строк с ключом key
* edit(): возвращает объект SharedPreferences.Editor, который используется для редактирования настроек.

Для управления настройками используется объект класса **SharedPreferences.Editor**, возвращаемый метод edit(). Он определяет следующие методы:

* clear(): удаляет все настройки
* remove(String key): удаляет из настроек значение с ключом key
* putBoolean(String key, boolean value): добавляет в настройки значение типа boolean с ключом key
* putFloat(String key, float value): добавляет в настройки значение типа float с ключом key
* putInt(String key, int value): добавляет в настройки значение int с ключом key
* putLong(String key, long value): добавляет в настройки значение типа long с ключом key
* putString(String key, String value): добавляет в настройки строку с ключом key
* putStringSet(String key, Set<String> values): добавляет в настройки строковый массив
* commit(): подтверждает все изменения в настройках
* apply(): также, как и метод commit(), подтверждает все изменения в настройках, однако измененный объект SharedPreferences вначале сохраняется во временной памяти, и лишь затем в результате асинхронной операции записывается на мобильное устройство.

Рассмотрим пример сохранения и получения настроек в приложении. Допустим, у нас следующий файл layout с пользовательским интерфейсом:

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

    xmlns:tools="<http://schemas.android.com/tools>"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical">

    <EditText

        android:id="@+id/nameBox"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:hint="Введите имя"/>

    <Button

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Сохранить"

        android:onClick="saveName"/>

    <TextView

        android:id="@+id/nameView"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:textSize="18dip"/>

    <Button

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Получить имя"

        android:onClick="getName"/>

</LinearLayout>

На экране будут две кнопки - для сохранения и для вывода ранее сохраненного значения, а также поле для ввода и текстовое поля для вывода сохраненной настройки.

Определим методы обработчики кнопок в классе Activity:

package com.example.eugene.preferencesapp;

import android.support.v7.app.ActionBarActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.TextView;

import android.widget.EditText;

import android.view.View;

import android.content.SharedPreferences;

public class MainActivity extends ActionBarActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

    }

    public void saveName(View view) {

        // получаем введенное имя

        EditText nameBox = (EditText) findViewById(R.id.nameBox);

        String name = nameBox.getText().toString();

        // сохраняем его в настройках

        SharedPreferences settingsActivity = getSharedPreferences("PersonalAccount", MODE\_PRIVATE);

        SharedPreferences.Editor prefEditor = settingsActivity.edit();

        prefEditor.putString("Name", name);

        prefEditor.commit();

    }

    public void getName(View view) {

        // получаем сохраненное имя

        TextView nameView = (TextView) findViewById(R.id.nameView);

        SharedPreferences settingsActivity = getSharedPreferences("PersonalAccount", MODE\_PRIVATE);

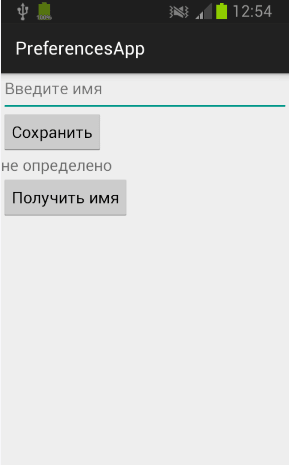
        String name = settingsActivity.getString("Name","не определено");

        nameView.setText(name);

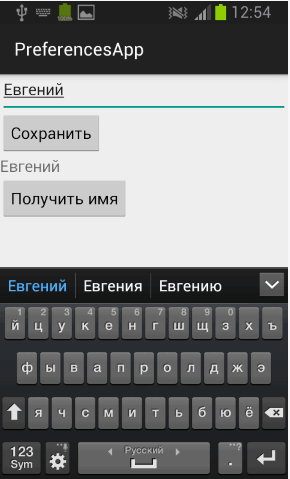
    }

}

При отсутствии настроек при попытке их получить, приложение выведет значение по умолчанию:



Теперь сохраним и выведем заново сохраненное значение:



**Приватные настройки**

Кроме общих настроек каждая activity может использовать приватные, к которым доступ из других activity будет невозможен. Для получения настроек уровня activity используется метод **getPreferences(MODE\_PRIVATE)**:

import android.content.SharedPreferences;

//........................

SharedPreferences settingsActivity = getPreferences(MODE\_PRIVATE);

То есть в отличие от общих настроек здесь не используется название группы настроек в качестве первого параметра, как в методе getSharedPreferences(). Однако вся остальная работа по добавлению, получению и изменению настроек будет аналогична работает с общими настройками.