Сохранение настроек - Флажки

Расширим функциональность блокнота, добавив в него возможность сохранения различных настроек.

Подготовим различные настройки, которые будем хранить в файле строковых ресурсов **strings.xml**:

res/values/strings.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="app\_name">Notepad</string>

<string name="action\_open">Открыть</string>

<string name="action\_save">Сохранить</string>

<string name="action\_settings">Настройки</string>

<string name="pref\_openmode">Режим открытия</string>

<string name="pref\_color">Цвет</string>

<string name="pref\_color\_black">Черный цвет</string>

<string name="pref\_color\_red">Красный цвет</string>

<string name="pref\_color\_green">Зеленый цвет</string>

<string name="pref\_color\_blue">Синий цвет</string>

<string name="pref\_size">Размер</string>

<string name="pref\_style">Стиль</string>

<string name="pref\_style\_regular">Нормальный</string>

<string name="pref\_style\_bold">Полужирный</string>

<string name="pref\_style\_italic">Курсив</string>

</resources>

Теперь создадим файл настроек **preferences.xml** (если забыли, то перечитайте статью [Сохранение настроек](http://developer.alexanderklimov.ru/android/preference.php)):

res/xml/preferences.xml

<?xml version="1.0" encoding="utf-8"?>

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

<CheckBoxPreference

android:key="@string/pref\_openmode"

android:summary="Открывать файл при запуске приложения"

android:title="Открыть" />

</PreferenceScreen>

Итак, мы определили первую настройку под именем **pref\_openmode**, которая будет или сразу загружать файл в поле редактирования, если установлен флажок, или открывать пустое поле, если флажок не установлен.

Создадим новую активность, которая наследует от класса **PreferenceActivity**. В классе активности для предпочтений внутри метода обратного вызова **onCreate()** нужно только вызвать метод **addPreferencesFromResource()** и загрузить XML-файл **preferences.xml**, содержащий наши настройки (пока одну):

SettingsActivity.java

package ru.alexanderklimov.notepad;

import android.os.Bundle;

import android.preference.PreferenceActivity;

public class SettingsActivity extends PreferenceActivity {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// загружаем предпочтения из ресурсов

addPreferencesFromResource(R.xml.preferences);

}

}

Не забываем добавить объявление активности **SettingsActivity** в файл манифеста AndroidManifest.xml:

<activity

android:name=".SettingsActivity"

android:label="@string/app\_name">

</activity>

Сейчас метод **addPreferencesFromResource()** считается устаревшим и для Android 4.0 и выше следует использовать другой код. Но ничего страшного.

Добавим новый пункт меню **Настройки** в меню (файл **res/menu/main.xml**), чтобы открывать подготовленное окно настроек.

<item

android:id="@+id/action\_settings"

android:orderInCategory="100"

android:showAsAction="never"

android:title="@string/action\_settings"/>

В методе обратного вызова **onOptionsItemSelected()** добавим новый блок **case** для оператора **switch**:

case R.id.action\_settings:

Intent intent = new Intent();

intent.setClass(this, SettingsActivity.class);

startActivity(intent);

return true;

Запустите приложение и убедитесь, что в меню появился новый пункт, который открывает окно настроек.

Чтение установок предпочтений нужно проводить в методе **onResume()**, который вызывается системой как во время запуска приложения, так и после закрытия окна настроек и возврата главной активности на передний план:

@Override

public void onResume() {

super.onResume();

SharedPreferences prefs = PreferenceManager

.getDefaultSharedPreferences(this);

// читаем установленное значение из CheckBoxPreference

if (prefs.getBoolean(getString(R.string.pref\_openmode), false)) {

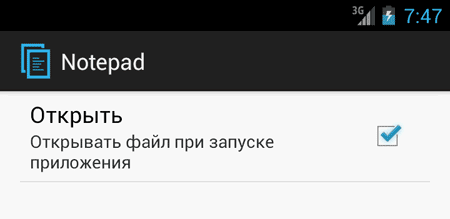
openFile(FILENAME);

}

}

В методе **getBoolean()** второй параметр *false* означает значение по умолчанию для возвращаемого значения предпочтения, если запрос на чтение установленного значения закончится неудачей.

Мы создали первую настройку, позволяющую сразу открывать нужный файл для редактирования, если отметить галочкой нужный пункт.



Сохранение текстовых настроек

Добавим возможность устанавливать размер шрифта для текста. Откроем снова файл **preferences.xml** и добавим новый элемент **EditTextPreference**:

<EditTextPreference

android:defaultValue="14"

android:dialogTitle="Введите размер шрифта (от 10 до 32)"

android:key="@string/pref\_size"

android:summary="Устанавливает новый размер шрифта"

android:title="Размер шрифта" />

В метод **onResume()** добавим новый код для чтения установленного значения размера шрифта:

// читаем размер шрифта из EditTextPreference

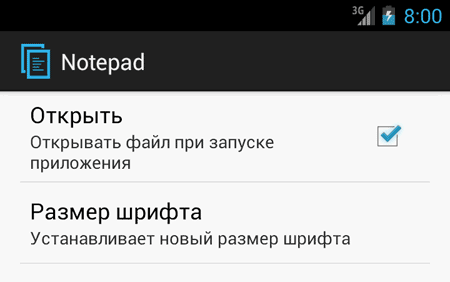
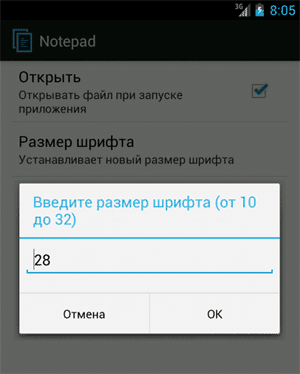
float fSize = Float.parseFloat(prefs.getString(

getString(R.string.pref\_size), "20"));

// применяем настройки в текстовом поле

mEditText.setTextSize(fSize);

Запустите проект и вызовите окно настроек. Теперь у нас появилась опция установки размера шрифта. Если щёлкнуть на новом пункте, то откроется диалоговое окно с текстовым полем ввода.

Обращаю ваше внимание, что в нашем примере не проверяется пользовательский ввод, что может привести к ошибкам, если вместо числового значения для размера шрифта пользователь введет слово **Кот** или любое другое слово из трёх букв. Никогда не доверяйте пользователю!

Сохранение настроек - Списки

Продолжим работу с текстовым редактором и добавим в него список для выбора стиля текста. В списке будет четыре опции: Обычный, Полужирный, Курсив, Полужирный+Курсив.

Подготовим массив строк и сохраним его в файле **arrays.xml**, который необходимо поместить в каталог **res/values/**.

res/values/arrays.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string-array name="text\_style">

<item>Обычный</item>

<item>Полужирный</item>

<item>Курсив</item>

<item>Полужирный+Курсив</item>

</string-array>

</resources>

В файл **preferences.xml** добавим дополнительный элемент <ListPreference>, в котором определим атрибуты заголовка окна, привязку к массиву значений и значение по умолчанию:

<ListPreference

android:key="@string/pref\_style"

android:title="Стиль для шрифта"

android:summary="Устанавливает стиль для шрифта"

android:defaultValue="1"

android:entries="@array/text\_style"

android:entryValues="@array/text\_style"

android:dialogTitle="Выберите стиль для шрифта" />

Для чтения настроек из списка добавляем код в метод **onResume()**:

// читаем стили текста из ListPreference

String regular = prefs.getString(getString(R.string.pref\_style), "");

int typeface = Typeface.NORMAL;

if (regular.contains("Полужирный"))

typeface += Typeface.BOLD;

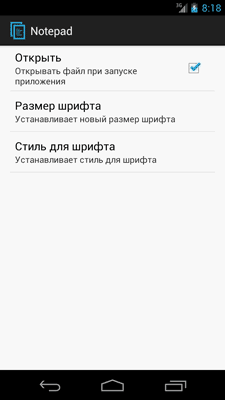
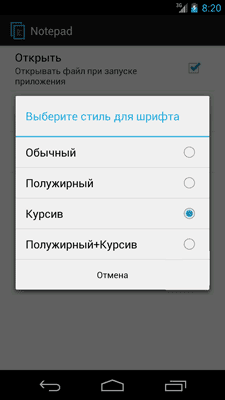
if (regular.contains("Курсив"))

typeface += Typeface.ITALIC;

// меняем настройки в EditText

mEditText.setTypeface(null, typeface);

Запустив проект, вы теперь увидите новую настройку **Стиль для шрифта**, которая открывает диалоговое окно для выбора стиля из списка. Обратите внимание, что в диалоговом окне нет кнопки сохранения, только **Отмена**. Изменения сохраняются сразу при выборе опции списка.

В статье [Сохранение настроек](http://developer.alexanderklimov.ru/android/preference.php#preferencecategory) можно прочитать дополнительные материалы по улучшению окна настроек. Например, можно добавить новую настройку, связанную с выбором цвета для текста.

