## Хранение данных в общих настройках

### Цель работы

Научиться сохранять данные и состояние приложения в файле общих настроек, освоить операции чтения и записи в них, восстанавливать состояние из сохраненного файла

### Задания для выполнения

Создайте приложение, единственная активность которого состоит из формы, в которую пользователь вводит следующую информацию о студенте: имя, группа (выбор одного элемента из нескольких заранее определенных), возраст (в виде ползунка), дата рождения (в виде поля ввода даты).

### Методические указания

Для примера разберем более простое приложение, состоящее из одного текстового поля ввода. Для начала создадим макет главной активности:

<EditText

android:id="@+id/etName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:ems="10"

android:inputType="textPersonName"

android:text=""

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"/>

<Button

android:id="@+id/btnSave"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Сохранить"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.174"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.717" />

<Button

android:id="@+id/btnLoad"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Восстановить"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.937"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.717" />

Теперь перейдем к коду. Для начала нам понадобится создать три поля класса, содержащие наши элементы управления, так как мы будем обращаться с ним в нескольких методах:

Button btnSave, btnLoad;

EditText etName;

Содержание метода onCreate будет полностью стандартным: привязка к макету, получение ссылок на визуальные элементы и задание обработчиков событий:

**btnSave** = findViewById(R.id.***btnSave***);

**btnLoad** = findViewById(R.id.***btnLoad***);

**etName** = findViewById(R.id.***etName***);

**btnSave**.setOnClickListener(**this**);

**btnLoad**.setOnClickListener(**this**);

Теперь нам осталось создать метод onClick. Автозаполнение студии поможет вам в этом и сделает все за вас автоматически. Как обычно, когда один обработчик события привязан к нескольким элементам, нам понадобится множественный выбор:

@Override

**public void** onClick(View v) {

String name = **""**;

**switch** (v.getId()){

**case** R.id.***btnSave***:

**break**;

**case** R.id.***btnLoad***:

**break**;

**default**:

**break**;

}

}

Теперь, когда весь вспомогательный код готов, можно приступить к содержательной части - работе с файлом настроек. Для начала заполним ветку сохранения данных. Первым делом нам надо получить ссылку на файл настроек и оттуда получить объект редактора этого файла. Это можно сделать в две строки:

SharedPreferences pref = getPreferences(MODE\_PRIVATE);

SharedPreferences.Editor ed = pref.edit();

А можно и в одну, для краткости:

SharedPreferences.Editor ed = getPreferences(MODE\_PRIVATE).edit();

Теперь нам осталось только получить данные из поля ввода и записать их в файл. В файле настроек данные хранятся в виде набора пар имя-значение. Поэтому для записи нам нужно придумать уникальное имя (или ключ) для нашей записи. Это будет первым параметром в методе putString, который и записывает данные в файл настроек. Обратите внимание на команду коммита в редакторе:

name = **etName**.getText().toString();

ed.putString(**"name"**, name);

ed.commit();

Log.*i*(**"SPREF"**, name);

Еще проще будет заполнение метода чтения данных из файла. Нам всего лишь потребуется опять получить ссылку на файл (получать редактор уже не нужно, ведь мы будем только читать), получить оттуда строку по ключу и записать ее в наше текстовое поле:

SharedPreferences pref = getPreferences(***MODE\_PRIVATE***);

name = pref.getString(**"name"**, **""**);

**etName**.setText(name);

Основная работа уже сделана. Теперь можно заняться рефакторингом кода. Предоставляем вам самостоятельно выделить всю сутевую часть из обработчиков событий в отдельные функции, добавить другие запоминаемые поля ввода и проверить работоспособность программы.

### Контрольные вопросы

1. Что такое общие настройки и зачем они нужны?

Общие настройки это некоторое хранилище из пар ключ-значение. В них можно сохранить некоторое состояние приложения, какие-либо настройки интерфеса, которые необходимо запоминать при выключении и применять при старте.

1. Чем отличаются приватные и публичные общие настройки?

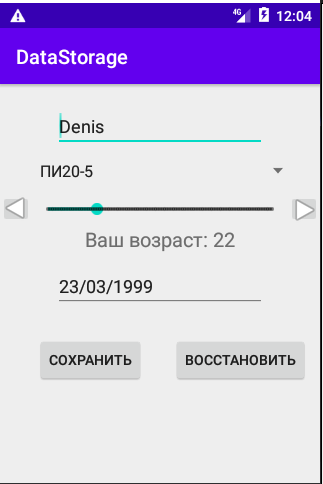
К приватным доступ может получить только активность – родитель, а к общим – любые.

1. Какие данные не целесообразно хранить в файле общих настроек?

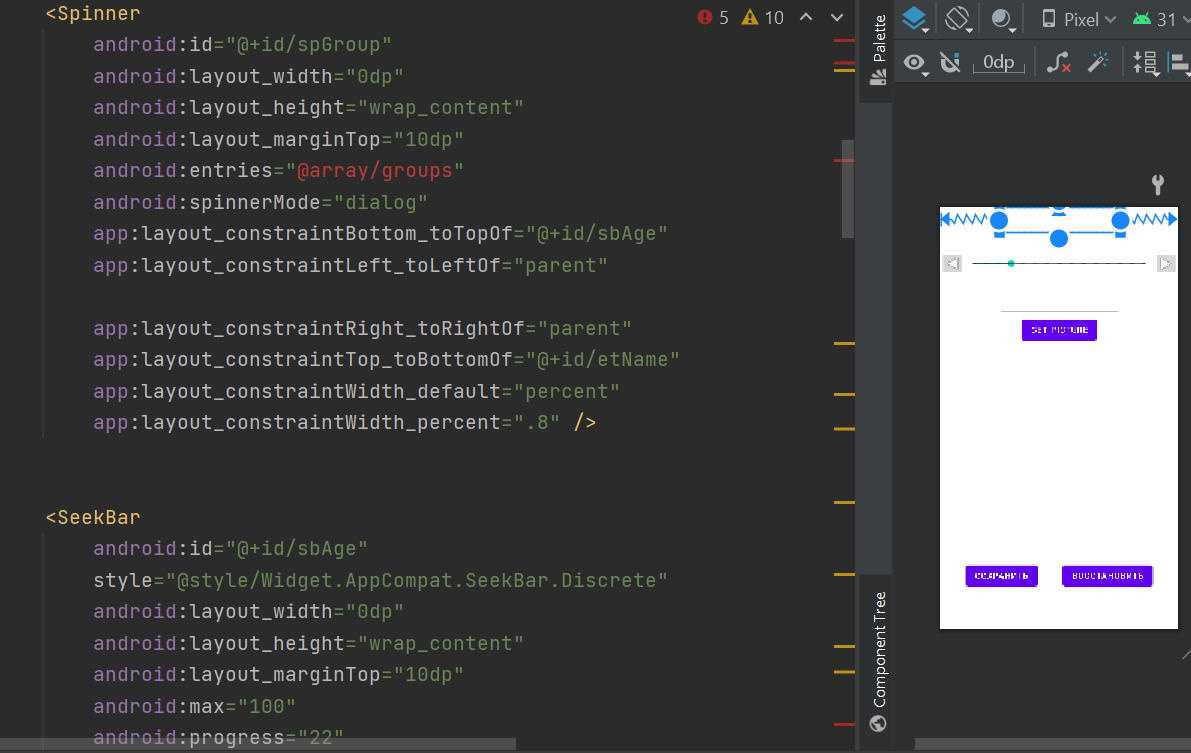
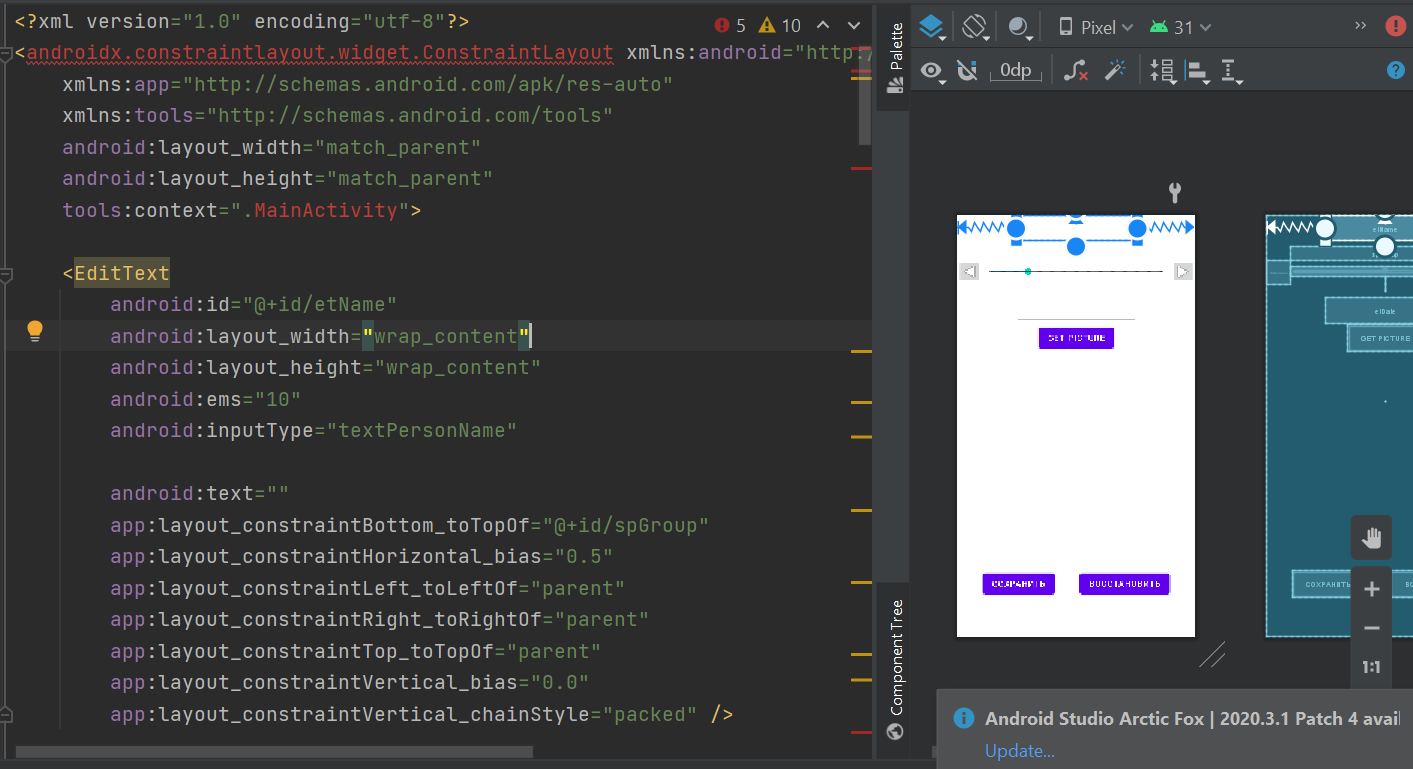
Не стоит хранить данные сложной структуры и больших объемов, больше конфигурационные, легкие, простые. Тем более не следует хранить важные с точки зрения безопасности данные, вроде паролей и т.п.

### Дополнительные задания

1. Модифицируйте программу таким образом, чтобы все необходимые данные автоматически сохранялись при остановке приложения и восстанавливались при возобновлении работы.



Activity\_main.xml

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Strings.xml

<resources>  
 <string name="app\_name">DataStorage</string>  
 <string-array name="groups">  
 <item>ПИ20-1</item>  
 <item>ПИ20-2</item>  
 <item>ПИ20-3</item>  
 <item>ПИ20-4</item>  
 <item>ПИ20-5</item>  
 <item>ПИ20-6</item>  
 </string-array>  
</resources>

MainACtivity.java

package com.nikita.datastorage;  
  
import android.content.ActivityNotFoundException;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.graphics.drawable.BitmapDrawable;  
import android.os.Environment;  
import android.provider.MediaStore;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.ImageView;  
import android.widget.SeekBar;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.OutputStream;  
import java.sql.Time;  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity implements View.OnClickListener, SeekBar.OnSeekBarChangeListener{  
  
  
 Button btnSave, btnLoad, btnImage;  
 EditText etName, etDate;  
 Spinner spGroup;  
 SeekBar sbAge;  
 TextView tvAge;  
 ImageButton ibDecrease, ibIncrease;  
 ImageView ivPicture;  
 ArrayList<String> groups = new ArrayList<String>();  
 String[] groupNames = new String[] {"ПИ", "ПМ", "ИБ", "БИ"};  
  
 private void savePicture(ImageView iv, String folderToSave, int id)  
 {  
 OutputStream fOut;  
  
 try {  
 File file = new File(folderToSave, "my\_image.jpg");  
  
 fOut = new FileOutputStream(file);  
 Bitmap bitmap = (Bitmap) ((BitmapDrawable) iv.getDrawable()).getBitmap();  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*, id, fOut);  
 fOut.flush();  
 fOut.close();  
  
// MediaStore.Images.Media.insertImage(getContentResolver(), file.getAbsolutePath(), file.getName(), file.getName());  
 }  
 catch (Exception e){  
  
 }  
  
 }  
  
 private void setInstanceState() {  
 int age,group;  
 String name, date;  
 SharedPreferences pref = getPreferences(*MODE\_PRIVATE*);  
 SharedPreferences.Editor ed = pref.edit();  
 name = etName.getText().toString();  
 group = spGroup.getSelectedItemPosition();  
 age = sbAge.getProgress();  
 date = etDate.getText().toString();  
  
 ed.putString("name", name);  
 ed.putInt("group", group);  
 ed.putInt("age", age);  
 ed.putString("birthday", date);  
  
 ed.commit();  
  
  
 String path = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*).toString();  
 savePicture(ivPicture, path, 100);  
 }  
  
 private void getInstanceState() {  
 int age,group;  
 String name, date;  
 SharedPreferences pref = getPreferences(*MODE\_PRIVATE*);  
 name = pref.getString("name", "Nikita");  
 Log.*i*("SPREF", name);  
 group = pref.getInt("group", 16);  
 Log.*i*("SPREF", Integer.*toString*(group));  
 age = pref.getInt("age", 22);  
 Log.*i*("SPREF", Integer.*toString*(age));  
 date = pref.getString("birthday", "13.03.1999");  
 Log.*i*("SPREF", date);  
  
 etName.setText(name);  
 spGroup.setSelection( group);  
 sbAge.setProgress(age);  
 etDate.setText(date);  
  
  
 String path = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*).toString() + "/my\_image.jpg";  
 File imgFile = new File(path);  
 if(imgFile.exists())  
 {  
 Bitmap myBitmap = BitmapFactory.*decodeFile*(imgFile.getAbsolutePath());  
 ImageView imageView=(ImageView)findViewById(R.id.*ivPicture*);  
 imageView.setImageBitmap(myBitmap);  
 }  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState){  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 for (String gr: groupNames)  
 for (int i = 18; i<=21; i++)  
 for (int j = 1; j<=6; j++)  
 groups.add(gr + Integer.*toString*(i) + "-" + Integer.*toString*(j));  
 btnSave = findViewById(R.id.*btnSave*);  
 btnLoad = findViewById(R.id.*btnLoad*);  
 etName = findViewById(R.id.*etName*);  
 etDate = findViewById(R.id.*etDate*);  
 spGroup = findViewById(R.id.*spGroup*);  
 tvAge = findViewById(R.id.*tvAge*);  
 sbAge = findViewById(R.id.*sbAge*);  
 ibDecrease = findViewById(R.id.*ibDecrease*);  
 ibIncrease = findViewById(R.id.*ibIncrease*);  
 ivPicture = findViewById(R.id.*ivPicture*);  
 btnImage = findViewById(R.id.*btnImage*);  
  
  
 tvAge.setText("Ваш возраст: " + String.*valueOf*(sbAge.getProgress()));  
  
 ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_spinner\_item*,  
 groups);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spGroup.setAdapter(adapter);  
  
 btnSave.setOnClickListener(this);  
 btnLoad.setOnClickListener(this);  
 btnImage.setOnClickListener(this);  
 ibDecrease.setOnClickListener(this);  
 ibIncrease.setOnClickListener(this);  
 sbAge.setOnSeekBarChangeListener(this);  
  
 getInstanceState();  
 }  
  
  
  
 @Override  
 public void onClick(View v) {  
  
  
  
 switch (v.getId()){  
 case R.id.*btnSave*:  
 setInstanceState();  
 break;  
 case R.id.*btnLoad*:  
 getInstanceState();  
 break;  
 case R.id.*btnImage*:  
 Intent takePhotoIntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
 try{  
 startActivityForResult(takePhotoIntent, 1);  
 }catch (ActivityNotFoundException e){  
 e.printStackTrace();  
 };  
 break;  
 case R.id.*ibDecrease*:  
 sbAge.incrementProgressBy(-1);  
 break;  
 case R.id.*ibIncrease*:  
 sbAge.incrementProgressBy(1);  
 break;  
 default:  
 break;  
 }  
 }  
  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int i, boolean b) {  
 tvAge.setText("Ваш возраст: " + String.*valueOf*(sbAge.getProgress()));  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
  
// @Override  
// protected void onDestroy()  
// {  
// setInstanceState();  
// super.onDestroy();  
//  
// }  
  
 @Override  
 protected void onPause()  
 {  
 setInstanceState();  
 super.onPause();  
  
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == 1 && resultCode == *RESULT\_OK*) {  
 Bundle extras = data.getExtras();  
 Bitmap thumbnailBitmap = (Bitmap) extras.get("data");  
 ivPicture.setImageBitmap(thumbnailBitmap);  
 }  
 }  
}

andoidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.nikita.datastorage">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.DataStorage">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
 />  
</manifest>