



《移动终端开发技术》 电子教案

第二单元 SharedPreferences 存储数据

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	SharedPreferences 存储数据	单元教学学时	4 课时
		在整体设计中的位置	第 14 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够熟练使用 SharedPreferences 存储与读取数据，实现记住密码功能。	1、要求学生掌握创建 SharedPreferences 对象的方法； 2、要求学生掌握使用 SharedPreferences 存储数据的方法； 3、要求学生掌握使用 SharedPreferences 读取数据的方法。	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、 难点	教学重点：使用 SharedPreferences 存储数据的步骤 教学难点：创建 SharedPreferences 对象的方法		
教学方法	采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。 运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。		
课前需掌握的知识 点	MODE_PRIVATE：指定该 SharedPreferences 中的数据只能被本应用程序读写； MODE_APPEND：该文件的内容可以追加； MODE_WORLD_READABLE：指定该 SharedPreferences 中的数据可以被其他应用程序读； MODE_WORLD_WRITEABLE：指定该 SharedPreferences 中的数据可以被其他应用程序读写。Intent 一般用于启动 Activity、启动服务、发送广播等，承担了 Android 应用程序三大核心组件相互间的通信功能。		
教学任务 分解	任务一、SharedPreferences 的用法 任务二、使用 SharedPreferences 存储和读取数据 任务三、实现记住密码功能		
教学 总结			



一、情景导入

文件存储，这种存储方式需要将数据存储到文件中，然后通过流读取出来。操作比较麻烦，为此 Android 提供了一个轻量级的存储类（SharedPreferences），主要用于存储一些应用程序的配置参数，例如用户名、密码、自定义参数的设置等。接下来将讲解 SharedPreferences 的使用。

二、课前测试

(1) 以下哪些方法可以得到 SharedPreferences 对象（ABC）。

- A. `getSharedPreferences("data", 0)`
- B. `getPreferences()`
- C. `PreferencesManager.getDefaultSharedPreferences()`
- D. `openFileOutput("data", Context.MODE_PRIVATE)`
(返回文件输出流)

(2) 调用 Editor 的 (B) 方法将添加的数据提交，从而完成数据存储操作。

- A. `edit()`
(SharedPreferences 的方法，获取编辑器返回 Editor 对象)
- B. `apply()`
- C. `putString("name", "Android")` (存入 String 类型数据)
- D. `putInt("age", 8);` (存入 int 类型数据)

(3) 获取 SharedPreferences 对象 sp 中键为 name 的字符串类型的值的语句是 (B)

- A. `sp.getInt("name", "");`
- B. `sp.getString("name", "");`
- C. `sp.getString("name", false);`
- D. `sp.getBoolean("name", false);`

(4) 下面 (B) 方法是 Activity 提供的专门用于启动活动的。

- A. `Intent(Context packageContext, Class<?> cls)`
(生成 Intent 的构造函数，接受两个参数，第一个 Context 要求提供一个启动活动的上下文，第二个 Class 为想要启动的目标活动)
- B. `startActivity(Intent intent)`
- C. `clear()` 清除内容
- D. `getPreferences()`

三、知识讲解

任务一、SharedPreferences 的使用

SharedPreferences 中存储的数据是以 key/value 键值对的形式保存在 XML 文件中，该文件位于“data/data/<packagename>/shared_prefs”文件夹中。需要注意的是，SharedPreferences 中的 value 值只能是 float、int、long、boolean、string、StringSet 类型数据。

使用 SharedPreferences 类存储数据时，首先要获取到 SharedPreferences 对象。Android 主要提供了 3 种方法得到 SharedPreferences 对象。

(1) 通过 Context 类的 `getSharedPreferences(String name,int mode)` 获取



SharedPreferences 的实例对象（在 Activity 中可以直接使用 this 代表上下文，如果不是在 Activity 中则需要传入一个 Context 对象获取上下文），示例代码如下：

```
SharedPreferences sp = context.getSharedPreferences(String name,int mode);
```

第二个参数目前只有 MODE_PRIVATE 这一个模式可选，它是默认的操作模式，和直接传入 0 的效果是相同的，指定该 SharedPreferences 中的数据只能被本应用程序读写。

（2）Activity 类中的 getPreferences()方法

这个方法和 Context 类的 getSharedPreferences(String name,int mode)相似，不过只接收一个操作模式的参数，因为使用这个方法会自动将当前活动的类名作为 SharedPreferences 的文件名。

（3）PreferencesManager 类中的 getDefaultSharedPreferences（）方法

这是一个静态方法，接收到一个 Context 参数，并自动使用当前应用程序的包名作为前缀来命名 SharedPreferences 文件。

得到了 SharedPreferences 对象之后，就可以开始向 SharedPreferences 文件存储数据了，主要可以分为 3 个步骤：

（1）调用 SharedPreferences 对象的 edit（）方法来获取一个 SharedPreferences.Editor 对象。

（2）向 SharedPreferences.Editor 添加数据，如添加布尔类型 putBoolean（）方法，字符串使用 putString（）方法。

（3）调用 apply（）方法将添加的数据提交，从而完成数据存储操作。

具体代码如下：

```
SharedPreferences sp = getSharedPreferences("data",MODE_PRIVATE);  
//data 表示文件名  
Editor editor = sp.edit();           // 获取编辑器  
editor.putString("name", "Android"); // 存入 String 类型数据  
editor.putInt("age", 8);             // 存入 int 类型数据  
editor.apply();                       // 提交修改
```

SharedPreferences 获取数据时比较简单，只需要创建 SharedPreferences 对象，然后使用该对象获取相应 key 的值即可，具体代码如下：

```
SharedPreferences sp = getSharedPreferences();  
String data = sp.getString("name",""); // 获取用户名
```

用 get 方法都接收 2 个参数，第一个参数是键，传入存储数据时使用的键就可以得到相应的值了，第二个参数是默认值，即表示当传入的键找不到对应的值会以什么样的默认值进行返回。整型设置 0，布尔类型设置 false。

任务二、使用 SharedPreferences 存储和读取数据

项目源码：

activity_main.xml

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



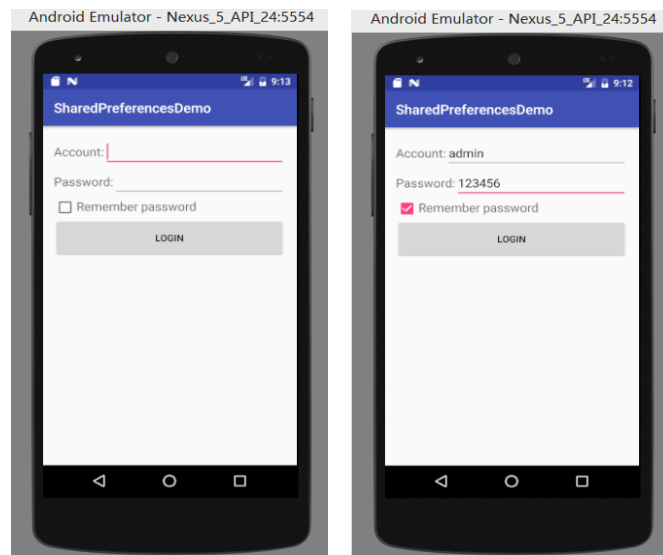
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical">
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button1"
    android:text="存储数据"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button2"
    android:text="读取数据" />
</LinearLayout>
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(final Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button saveData=(Button)findViewById(R.id.button1);
        Button restoreDate=(Button)findViewById(R.id.button2);
        saveData.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View view) {
                SharedPreferences
sharedPreferences=getSharedPreferences("data", MODE_PRIVATE);
                //第一步, 创建 SharedPreferences 对象
                Editor editor=sharedPreferences.edit();
                //第二步, 创建 Editor 对象
                editor.putString("name", "Tom");
                editor.putInt("age", 28);
                editor.putBoolean("married", false);
                //第三步, 向 Editor 对象中添加数据
                editor.apply();
                //第四步, 提交数据, 完成操作
            }
        });
        restoreDate.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View view) {
                SharedPreferences
```

```
sharedPreferences=getSharedPreferences("data", MODE_PRIVATE);  
    String name=sharedPreferences.getString("name","");  
    int age=sharedPreferences.getInt("age",0);  
    Boolean married=sharedPreferences.getBoolean("married",false);  
    Log.d("MainActivity","name is"+name);  
    Log.d("MainActivity","age is"+age);  
    Log.d("MainActivity","married is"+married);  
    }  
    });  
}
```

```
D/MainActivity: name isTom  
D/MainActivity: age is28  
D/MainActivity: married isfalse
```



任务三、实现记住密码的功能

项目源码:

activity_login.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:id="@+id/activity_login"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"  
    android:orientation="vertical">  
<LinearLayout
```



```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Account:"
    android:textSize="18sp"/>
<EditText
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:id="@+id/account"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Password:"
    android:textSize="18sp"/>
<EditText
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:id="@+id/password"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/remember_pass"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:text="Remember password"/>
</LinearLayout>
<Button
```



```
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:id="@+id/login"
        android:text="login"/>
</LinearLayout>

LoginActivity.java

    public class LoginActivity extends AppCompatActivity {
    private SharedPreferences pref;
    private SharedPreferences.Editor editor;
    private EditText accountEdit;
    private EditText passwordEdit;
    private CheckBox rememberPass;
    private Button login;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        pref= PreferenceManager.getDefaultSharedPreferences(this);
        accountEdit=(EditText)findViewById(R.id.account);
        passwordEdit=(EditText)findViewById(R.id.password);
        rememberPass=(CheckBox)findViewById(R.id.remember_pass);
        login=(Button)findViewById(R.id.login);
        Boolean isRemember=pref.getBoolean("remember_password", false);
        if(isRemember){
            //将账号和密码设置到文本框中
            String account=pref.getString("account","");
            String password=pref.getString("password","");
            accountEdit.setText(account);
            passwordEdit.setText(password);
            rememberPass.setChecked(true);
        }

        login.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View view) {
                String account=accountEdit.getText().toString();
                String password=passwordEdit.getText().toString();
                //如果账号为 admin, 密码为 123456, 则登录成功
                if(account.equals("admin")&&password.equals("123456")){
```




```
        editor=pref.edit();
        if(rememberPass.isChecked()){
            editor.putString("account",account);
            editor.putString("password",password);
            editor.putBoolean("remember_password",true);
        }
        else
            editor.clear();
        editor.apply();
        Intent intent=new
Intent(LoginActivity.this,MainActivity.class);
        startActivity(intent);
        finish();
    }
    else {
        Toast.makeText(LoginActivity.this,"account or password is
invalid",Toast.LENGTH_LONG).show();
    }
}
});
}
```

四、演示作品

各小组演示汇报作品，教师与其他小组总结评价作品，各小组课后完善作品并提交到教学平台。

五、知识巩固

- 1、总结知识点，使用教学平台中的随堂练习题巩固本所学知识。
- 2、使用教学平台中的测试题给学生布置作业。

拓展
作业

- 1、预习微课《创建数据库》
- 2、完成实现记住密码功能

教学
后记