بسم الله الرحمن الرحیم  
تو این قسمت قصد داریم کلاسی تحت عنوان setting بسازیم که برای ذخیره کردن نام کاربر در حافظه کش اپ که کاربر هر موقع که برنامه را باز میکند نیاز نباشد ثبت نام کند  
پس اول از هرکاری یک کلاس جدید میسازیم  
برای ساخت کلاس روی پکیج یا قسمتی که می خواهیم کلاس را در انجا بسازیم در پکیج مادر ان راست کلیک میکنیم و بعد new/java class بعد نام کلاس را وارد میکنیم که ما نام کلاس را setting میگذاریم  
  
توجه  
و برای استفاده از sharedpreferences در کلاس setting باید sharedpreferences Context , را باید import کنیم

نکته sharedpreferences چی هستش اصلا ؟

**به ما اجازه میدهد تا اطلاعات را با فرمت Key/Value (کلید/مقدار) ذخیره و نگهداری کنیم. این اطلاعات با فرمت xml ذخیره شده و تا زمانی که اپلیکیشن از روی سیستم عامل حذف نشده و یا کاربر به صورت دستی در قسمت تنظیمات برنامه ها، Data ی اپلیکیشن را حذف نکند، باقی می ماند**

**متد ها هم**

getPreferences()

getSharedPreferences()

getDefaultSharedPreferences()

به این صورت در اول کد های کلاس می نویسم

import android.content.Context;  
import android.content.SharedPreferences;

خب ما برای شروع به چند متغیر نیاز داریم که چند مقدار در ان ذخیره کنیم

private static String *TITLE*="myappsetting";  
private static String *FULLNNAME*="fullname";  
private static String *NO\_NAME*="no name yet";

توجه  
متغیر هارا از نوع static تعریف کردیم چرا ؟ چون می خواهیم در کلاس دیگری به ان ها دسترسی داشته باشیم

متغیر اولی که به نام TITLE میباشد برای ذخیره کردن نام فایل sharedpreferences میباشید که مقدار ان myappsetting است

متغیر دومی که به نام FULLNAME میباشد که مقدار key را در ان مقدار دهی میکنیم   
  
توجه روند ذخیره سازی مقادیر در sharedpreferences به صورت KEY-Value میباشد یعنی هر مقدار یک key دارید که با key میشود به ان دسترسی پیدا کرد   
  
متغیر دومی که به نام NO\_NAME میباشد که مقدار Defualt را در ان مقدار دهی میکنیم یعنی اگر اسمی در حافظه ذخیره نشده بود این مقدار را به ما بدهد البته توجه کنید در حال حاضر این قاعده ای است که ما در نظر گرفته ای ایم و هنوز از متغیر ها استفاده نکرده ایم

و بعد باید 2 متد بنویسیم  
   
یکی برای ذخیره کردن مقدار نام در sharedpreferences

و یکی نیز برای خواندن نام کاربر از sharedpreferences  
  
خب متد اول را با نام " Readpreferences " را با خروجی String که نام کاربر را برمیگرداند می نویسیم توجه فرمایید که این متد هم باید به صورت static تعریف شود چرا ؟ چون می خواهیم بدون instance

گرفتن از کلاس setting به انها دسترسی داشته باشیم دیگر نیاز به instance گرفتن از انها نشود

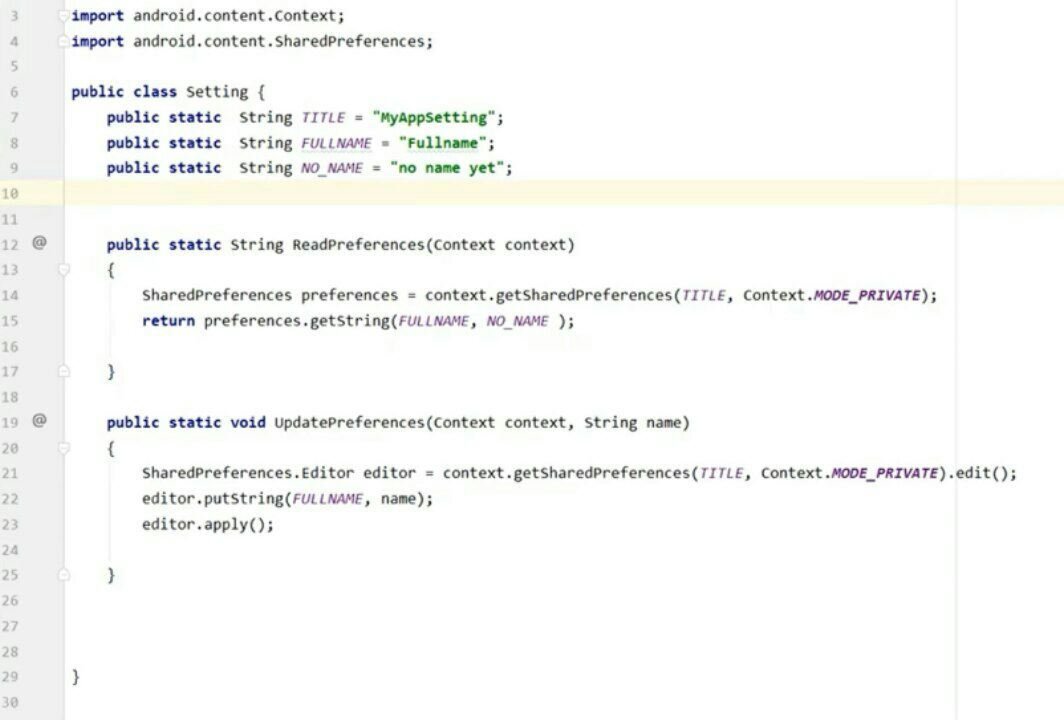
public static String ReadPreferences(Context context){  
 SharedPreferences SharedPreferences=context.getSharedPreferences(*TITLE*,Context.*MODE\_PRIVATE*);  
 return SharedPreferences.getString(*FULLNNAME*,*NO\_NAME*);  
}

در خط اول متد امدیم یک ابجکت از کلاس SharedPerferncess ساختیم و برای دسترسی به SharedPerferncess نیاز به context داریم همون طور که می بینید در پارامتر این متد ما context رو میگیریم و با متد getSharedPerfrencess از context میایم یک ابجکت از شیرید پرفرنس میگیرم که ۲ پارامتر دارد در پارامتر اول نام این SharedPerferncess رو وارد میکنیم که ما در متغیر TITLE نوشته ایم و در پارامتر دوم مد رو انتخاب میکنیم که همان طور که در عکس میبینید مد خصوصی(private) را داده ایم و همان طور که میبینید این متد String برمیگرداند پس ما با استفاده از ابجکتی که از شیرید پرفرنس گرفتیم اطلاعات مون رو ریتر یا همان برمیگردانیم با استفاده از متد getString که در پارامتر اولش key رو میگیره و در پارامتر دوم مقدار دیفالت که یعنی اگر دیتایی وجود نداشت چه برگرداند  
  
  
خب حالا میریم به سراغ ساخت متد updatePerfrences  
  
می توان گفت کار اصلیمون این قسمت که مقداری رو ذخیره کنیم

public static void updatepPreferences(Context context,String name){  
SharedPreferences.Editor editor=context.getSharedPreferences(*TITLE*,Context.*MODE\_PRIVATE*).edit();  
editor.putString(*FULLNNAME*,name);  
editor.apply();  
}

متد updateperfrencescs که کار اصلی این متد ذخیره کردن یک مقدار در شیرید پرفرنسس میباشد مثل توضیحات قبلی در اینجا هم یک ابجکت از شیرید پرفرنسس میگیریم با این تفاوت که edit را به کلاس اضافه میکنیم که اینبار بتوانیم اطلاعات را نیز ذخیره بکنیم در خط دوم کد این متد با استفاده از متد putstring دیتایی از نوع رشته را ذخیره میکنیم در پارامتر اول key را وارد میکنیم که متغیر ان را در کلاس به نام FULNAME ساخته ایم و پارامتر دوم مقدار رشته را که می خواهیم ذخیره شود را میدهیم که در پارامتر متد گرفته ایم و در خط سوم با کال کردن متد apply دیتارو ذخیره میکنیم

پس الان کل کلاس setting مثل تصویر زیر شد



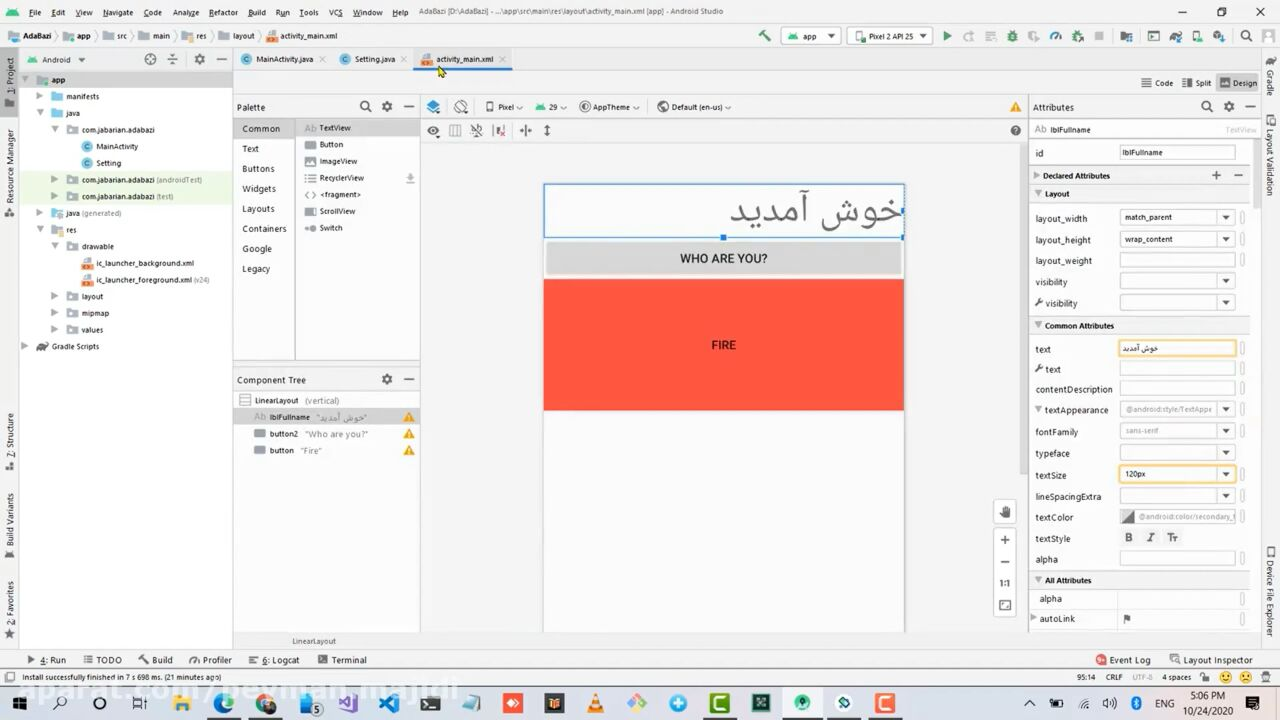
پس کلاس setting که کار اصلی این کلاس برای ذخیره کردن و خواندن نام کاربر ثبت نام کرده است تمام شد حال به سراغ استفاده از این کلاس و نمایش ان در صفحه Activity می رویم

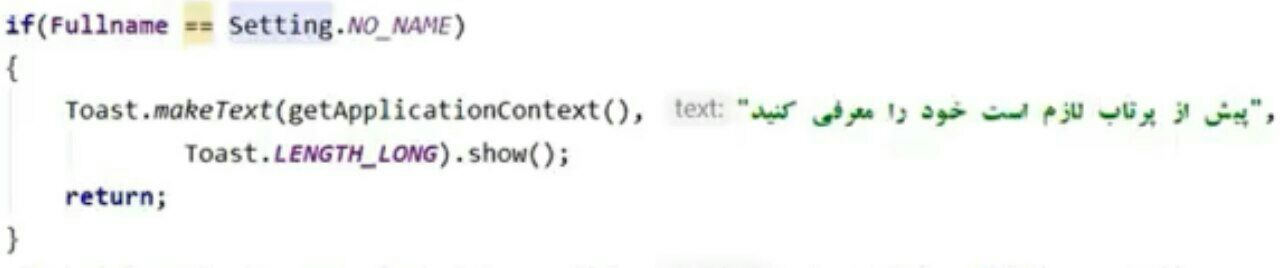
خب حال اول باید یک TextView به layout اکتیویتی اضافه کنیم تا نام کاربر را در انجا نمایش بدهیم و اگر کاربر ثبت نام نکرده بود متن " خوش امدید " را نمایش دهد  
  
برای اضافه کردن TextView باید از پنجره palette درگ کردن و به صفحه اکتیویتی انداخت و یا روش دیگر ان اضافه کردن از کد های XML میباشد

خب حال TextView را اضافه کردیم و می خواهیم یک سری مشخصاتی را به ان اضافه بکنیم attribiuts اندازه متن ((TextSize را به 120 px تغییر میدهیم

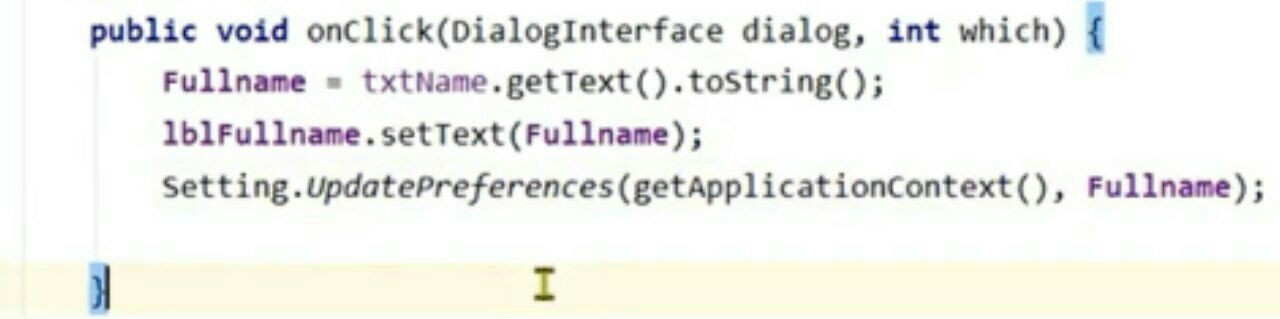
و یک id به TextView میدهیم   
  
دلیل استفاده از id در View ها چیست ؟

به دلیل این که ما با id ها می توانیم در logic برنامه به View دسترسی داشته باشیم به اصطلاح در واقعیت findviewByid میکنیم



خب حال دیزاین اکتیویتی ما شد تصویر بالا   
  
خب حال به قسمت کد های ارسال موشک می رویم که یک شرطی را بنویسیم که اگر کاربر ثبت نام نکرده بود نتواند موشک را ارسال کند  
  
در این قسمت که رویداد کلیک دکمه پرتاب موشک میباشد شرط میگذاریم که اگر Fulname برابر مقدار متغیرno\_NAME کلاس Setting بود در خط بعدی داخل شرط می گوییم به کاربر یک توست نشون بده که پیغام کوچکی است که پیش از پرتاب ثبت نام کنید و بعد return میکنیم وقتی retrun رو فرخوانی میکنیم دیگر از اینجا به بعد کد ها کار نمیکنید و اجرا نمیشود

خب حال به قسمت رویداد (Evant ) کلیک دیالوگ ثبت میرسیم   
که کد هارا به این صورت می نویسیم



در خط اول متن تکست ویویی که نام را وارد میکنیم را به تکست فول نیم میدهد و در خط بعد به تکست ویویی که قرار نام رو نشون بده متن رو ست میکنیم با fullname و در خط سوم از کلاس Setting استفاده میکنیم و با متد updatePrefrerences میدهیم که کد های ان را توضیح دادیم در تصاویر قبلی