کلاس های دیتا یکی از قدرتمندترین کلاس های کاتلین هستن که از نوشتن کدهای اضافی و بیهوده جلوگیری میکنن.

POJO : کلاس هایی که برای نگه داشتن وضعیت به کار میرن و معمولا فعالیت های ساده­ای رو انجام میدن. معمولا تنها توابع getter و setter ای مهیا میکنن که اجازه دسترسی به field هاشون رو میده.

تعریف کلاس های دیتا در کاتلین واقعا به سادگی کد زیره :

data class Forecast(val date: Date, val temperature: Float, val details: String)

**توابع اضافی:**

کلاس های دیتا یکسری توابع جالب در اختیارمون قرار میده و اینها همه جدا از دسترسی به خصیصه­هاش هستن.

* equals() : خصیصه­های دو شی رو با هم مقایسه میکنه که ببینه برابر هستن یا نه.
* hashCode() : hash-Code محاسبه شده از روی خصیصه هارو برمیگردونه.
* copy() : میتونیم با استفاده از این یک شی رو به علاوه با خصیصه هاش کپی کنیم
* تعدادی دیگه از توابع

**کپی کردن یک کلاس دیتا:**

اگه ما از خاصیت مصون بودن استفاده کنیم نمیتونیم مقدارش رو عوض کنیم، مگر این که یک مقدار جدید بهش بدیم( منظور اینه که یک شی جدید ازش بسازین و بهش نسبت بدین) . توی کلاس های دیتا تابعی ظاهر شده به نام copy() که کارش کپی کردن از یک شی­هه ولی یکی دیگه از خوبیاش اینه که شما هم میتونین از یک شی کپی بگیرین و هم مقدار جدید به جای خصیصه های دلخواهتون بذارین:

val *f1* = Forecast(Date(), 27.5f, "Shiny day")  
val *f2* = *f1*.copy(temperature = 30f)

الا ما با استفاده از این تابع دیتای شی اول رو کپی کردیم و تنها مقدار temperature رو عوض کردیم، بدون این که مقدار بقیه حالت ها و شی اول عوض شه !

**Map کردن یک شی به یک متغیر:**

این روند به نام multi-declaration انجام میگیره و به معنی مپ کردن هر خصیصه داخل یک شی به یک متغیره. (اگر دقت کرده باشین، در هنگام استفاده از شی ها، توابعی به نام ComponentX وجود دارن که به صورت اتومات ساخته میشن و دلیل ساخته شدنشون هم همینه). به مثال زیر دقت کنین:

val f1 = Forecast(Date(), 27.5f, "Shiny day")  
val (date, temperature, details) = f1

این multi-declaration به کد های زیر کامپایل میشه:

val date = f1.component1()  
val temperature = f1.component2()  
val details = f1.component3()

ایده پشت این ویژگی خیلی قدرتمنده و میتونه توی خلاصه­سازی کدمون توی شرایط مختلف کمک کنه. به عنوان مثال کلاس Map یکسری توابع الحاقی داره که پیاده سازی شده تا بتونن کلیدها و مقادیر اون هارو شناسایی کنن:

for ((key, value) in map) {  
 Log.d("map", "key:$key, value:$value")  
}