Batch#12/Android Class

Remote Learning Assignment-Week3

1. Encapsulation (封裝) 是將物件的狀態資訊隱藏在物件的內部，不允許外部程式直接訪問物件內部資訊，而是通過該類所提供的方法來實現對內部訊息的操作和訪問。

Benefits of Encapsulation:

* 讓使用者只能通過事先預訂的方法來訪問數據，因而可以在該方法裡加入控制邏輯，限制對屬性的不合理訪問。
* 可以進行數據的檢查，因而有利於保證物件訊息的完整性
* 方便修改，提高code的可維護性

1. enum class(枚舉類別)是一種特殊的類，它可以有自己的屬性、方法，可以實現一個或多個接口 (API)，也可以定義自己的建構子(constructor)。Enum類別可以幫變數指定對它來說有效的值(建立一組特定值)。Enum類別的每一個值都是一個常數，每一個常數都是一個enum類的實例(instance)。
2. enum class Season{

SRPING, SUMMER, FALL, WINTER

}

fun main (args : Array <String> ) {

for (s in Season . values () ) {

println(s)

}

val seasonName =” SUMMER”

val s: Season = Season.valusOf (seasonName)

println(s)

println(Season.WINTER)

}

1. LinearLayout和RelativeLayout兩者性能差異:
   * RelativeLayout慢於LinearLayout是因為它會讓子View調用2次measure過程，而LinearLayout只需要一次，但是有weight屬性存在時，LinearLayout也需要兩次measure。
   * RelativeLayout的子View如果高度和RelativeLayout不同，會導致RelativeLayout在onMeasure() 方法中做横向測量時，縱向的測量结果尚未完成，只好暫時使用自己的高度(myHeight)傳入子View系统。這樣子View的高度和RelativeLayout的高度不相同時，優化就會失敗。
   * 在不影響層級深度的情下(層級固定時)，使用LinearLayout而不是RelativeLayout。但如果層級多時使用Linearlayout。

詳細補述:

LinearLayout和RelativeLayout都屬於viewgroup。View的繪製過程分為三個部分: 測量(measure)、佈局 (Layout)和繪製(draw)。

View的繪製流程是從ViewRoot的performTraversals() 方法開始，依序經過measure()，layout()和draw()三個過程，才最終將一個View繪製出來。從ViewRoot的performTraversals()方法開始依次呼叫perfromMeasure、performLayout和performDraw這三個方法。這三個方法分別完成頂級View的measure、layout和draw三大流程，其中perfromMeasure會呼叫measure，measure又會呼叫onMeasure，在onMeasure方法中則會對所有子元素進行measure，這個時候measure流程就從父容器傳遞到子元素中了，這樣就完成了一次measure過程，接著子元素會重複父容器的measure，如此反覆就完成了整個View樹的遍歷。

首先RelativeLayout中子View的排列方式是基於彼此的依賴關係，而這個依賴關係可能和佈局中View的順序並不相同，在確定每個子View的位置的時候，需要先給所有的子View排序一下。又因為RelativeLayout允許A，B 2個子View，橫向上B依賴A，縱向上A依賴B。所以需要橫向縱向分別進行一次排序測量。

LinearLayout首先會對所有的子View進行measure，並計算totalWeight(所有子View的weight屬性之和) ，然後判斷子View的weight屬性是否為最大，如為最大則將剩餘的空間分配給它。如果不使用weight屬性進行佈局，則不進行第二次measure。

1. ConstraintLayout(約束布局)的優點是當佈局出現多層巢狀的時候，使用ConstraintLayout可以減少佈局巢狀，使用LinearLayout和RelativeLayout情況下，一層LinearLayout巢狀會導致onMeasure測量兩次，而RelativeLayout會測量四次，加上ConstraintLayout也向下相容到API level 9，所以ConstraintLayout能夠降低頁面佈局層級，提升專案的效能。