Batch #14 / Android Class Assignment - Week 3

1. **What is Encapsulation? Try to list out the benefits of Encapsulation.**

   Encapsulation 封裝是 object-oriented language 的一個核心優勢，類似於一個物件包裹的概念，這個設計可以將一組變數資料或函式方法打包到相同的單位當中，產生一個物件實體，最常見的封裝使用單位就是 class。封裝有個重要的功能是 Data Hiding，利用權限管控的修飾子，一般稱作 Access level modifer，可以限制其他組件訪問的機制，搭配繼承的設計讓約束更加彈性。在 Kotlin 中，此修飾稱為 Visibility modifiers，對資料做可見性控制，共有 private, protected , public(預設), internal 此 4 種等級。因為適當的隱藏數據，可避免數據發生嚴重的改動錯誤，一般對於類別變數常使用 private 讓其有限度的保護，可以另外宣告 Getter 和 Setter 方法(在 Kotlin 中系統自動創建)，並在其中約束對資料的行動，因此也可以讓資料唯讀或唯寫。除此之外，封裝增加程式代碼的重複運用，讀寫較為直覺，也方便系統的維護運營。

2. **Try to explain what enum is and why we use them.**

   列舉 Enum 是一個簡單對應名稱與資料的類型，列出需求範圍所有成員的設計，可以使用在一些固定範圍或希望被限制的資料，一般來說相同的列舉內容不會重複出現，每個列舉的內容都會包含 name 和 ordinal 屬性。列舉可以限制之後程式可以採用的值，因為常數設置的功能，也方便在編譯時期預做檢查，列舉同樣可和 generic, loop, Iterator 做搭配使用。

3. **How to use enum in Kotlin? Attach a sample code that uses enum in Kotlin.**

```kotlin
enum class Season(val plant: String, val feel: String){
    //kotlin將列舉實作在class
    SPRING("蘭", "Warm"),
    SUMMER("竹", "Hot"),
    AUTUMN("菊", "Cool"),
    WINTER("梅", "Cold")
}

fun main() {
    println("Season-${Season.SUMMER.name}, Plant-${Season.SUMMER.plant}")
    println("${Season.WINTER.ordinal}, Feel-${Season.WINTER.feel}")
}
```

```
Season-SUMMER, Plant-竹
3, Feel-Cold
```

4. **What are the differences between LinearLayout and RelativeLayout? Try to explain in detail.**

   一個 LinerLayout 必須在編程決定其方向性(android:orientation) Vertical, Horizontal，在 Layout 之下每一行(列)都只能置放一個子元素，他們會對齊 Layout 的邊界(寬/高)逐個堆疊。可以使用 LinerLayout 元素中的權重 layout_weight，分配既有的布局空間。RelativeLayout 的子元素可以指定他們相對於其他元素或 parent 的關聯位置，可以簡潔適應不同解析度或螢幕大小的設備。此兩種佈局可有千秋，主要抉擇於版面配置需要依順序和結構對齊排列，或是更著重於 Layout 和元素之間的對應位置。

5. **Try to explain the benefits of ConstraintLayout. Why should you use it?**

   ConstraintLayout 的位置與大小是由 layout_width, layout_height 以及元素間的 constraint 所互相約束，結合了其他 layout 的優點，既可以依順序堆疊，也能依相對位置要求做規劃。ConstraintLayout 提供的 chain 功能，有以下幾種模式 Spread, Spread inside, Packed 或 Weighted 承襲了權重的概念，可以把鏈結的幾個元素做空間上的分配調整。另外，因應現在各種介面或文件內容，響應式設計更顯友善。當我們想做到同樣的結果時，ConstraintLayout 減少許多繁瑣的巢狀 layout，在元素的調整上更加靈活。