

Week 3

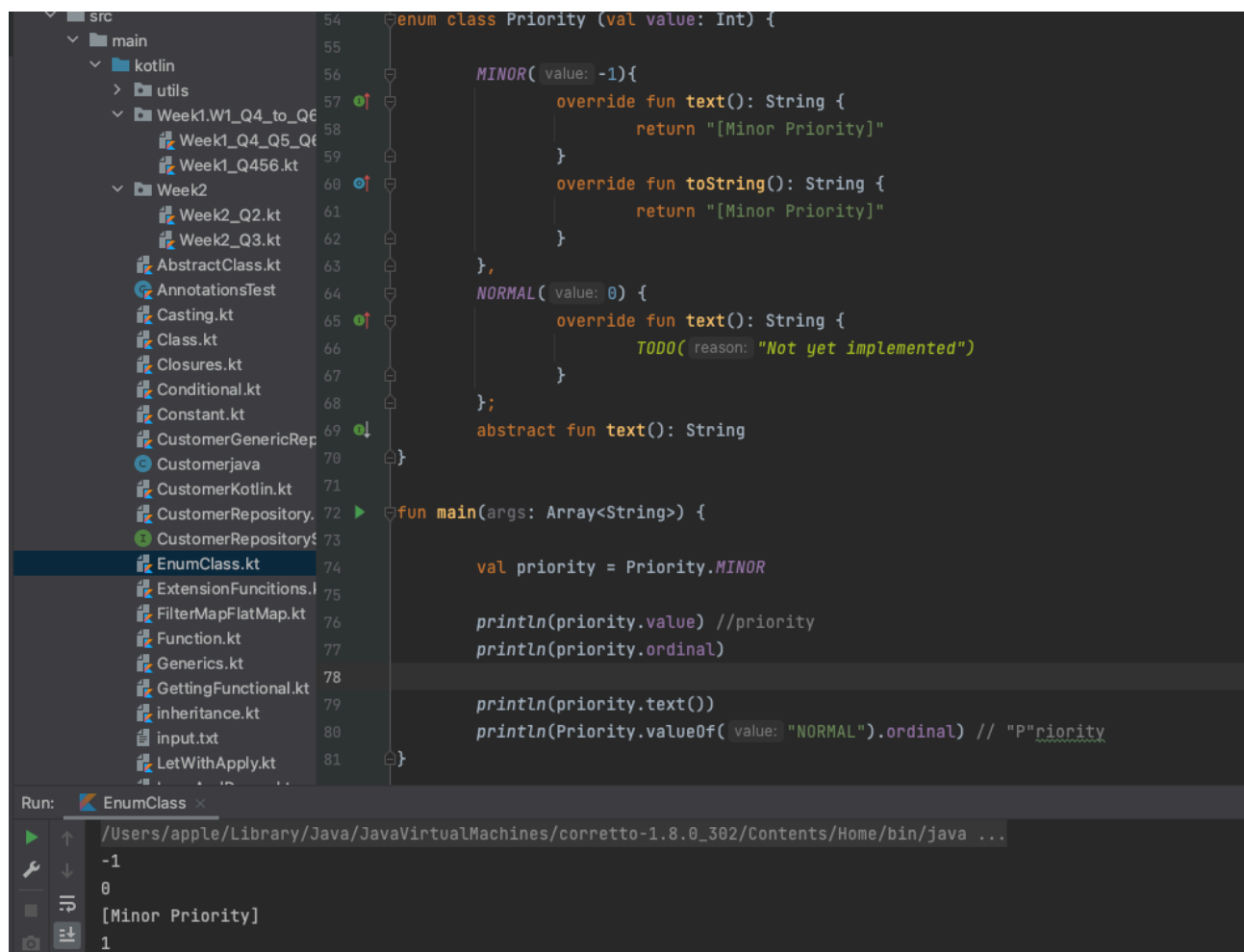
1. Encapsulation ?

ANS: 封裝，我們可以將特定class或function透過封裝的方式不讓外部任意更改它，外界只能從介面看到它實作結果，這個方式可以避免引入更多錯誤；像影片中在決定score這個值時，就是用封裝的方式，其值只能在ViewMode中實作產生，而不能在Fragment直接任意修改它，Fragment本身不應該進行數據的修改。

2. enum ?

ANS: 列舉，列舉類型可以將一組指定常數進行定義，並在此列舉中針對這些指定常數進行特定值的設定，列舉可以清楚簡單的呈現常數，可提高程式可讀性。

3. 要在Kotlin中使用列舉類型，可以在class前加上enum，其中，enum有內建的屬性以及方法，常數.name屬性代表列舉常數的名稱，列舉類型.values()方法可以呈現此class內的列舉常數有哪些，列舉類型.valueOf("列舉常數名").屬性則可以呈現針對此特定列舉常數名的屬性，例如.ordinal就代表呈現此列舉常數名的順序；另外，我們也可以針對方法進行override，例如針對.text()方法的覆寫可以呈現指定text：



```
54 enum class Priority (val value: Int) {
55
56     MINOR( value: -1){
57         override fun text(): String {
58             return "[Minor Priority]"
59         }
60         override fun toString(): String {
61             return "[Minor Priority]"
62         }
63     },
64     NORMAL( value: 0) {
65         override fun text(): String {
66             TODO( reason: "Not yet implemented")
67         }
68     };
69     abstract fun text(): String
70 }
71
72 fun main(args: Array<String>) {
73
74     val priority = Priority.MINOR
75
76     println(priority.value) //priority
77     println(priority.ordinal)
78
79     println(priority.text())
80     println(Priority.valueOf( value: "NORMAL").ordinal) // "P"riority
81 }
```

Run: EnumClass x

```
/Users/apple/Library/Java/JavaVirtualMachines/corretto-1.8.0_302/Contents/Home/bin/java ...
-1
0
[Minor Priority]
1
```

4. What are the differences between LinearLayout and RelativeLayout ?

ANS: LinearLayout不管垂直或水平，每一個child排列都要是呈現一直線；RelativeLayout的話在水平或垂直上，child可以被作為排列的parent來作為對齊依據。

5. ConstraintLayout ?

ANS: 透過LinearLayout與RelativeLayout只定義了child之間的對齊方式，如果要指定與畫面邊界的對齊距離則需要透過ConstraintLayout設定，ConstraintLayout在畫面的旋轉呈現上很重要。