Q1. What is Inheritance? Try to explain in Mandarin

A1.

透過繼承的方式，可以在某一個類別的基礎上，建立新的子類別，並同時具備原類別所開放的屬性與方法。在父類別(原類別)中寫入的屬性或方法可做為共用或被覆寫，若是共用，各自子類別中就不用再重覆寫相同的程式碼；若是父類別有不符合的功能卻想要用同樣的方法名稱，可以覆寫父類別的方法，改寫其方法的內容。而子類別中，也可以根據當下的功能需求，另外定義各自的方法。

使用方式是子類別透過關鍵字冒號 ”：” 來繼承父類別，要注意的是子類別要繼承父類別的話，父類別的修飾子必須加上open的關鍵字，才可以被繼承，若子類別要覆寫方法，父類別的方法也必須加上open關鍵字

Q2. How to split a string in Android? Explain your understanding, and even better to attach some sample code.

A2.

將分割的引數 (字元) 帶入 split 中，字串分割後會返回一個字串集合(List<String>)

Exp1 :

val str = "This is a string"

val split = str.split(' ') //用空格分割

println(split) // **[This, is, a, string]**

Exp2:

val menuData = "shandy,Dragon's Breath,5.91"

val data = menuData.split(',') //用逗號分割

println(data) // **[shandy, Dragon's Breath, 5.91]**

Q3. What kind of key-value pair collection do we use in Android? Attach a sample code that uses for-loop to print each key-value pair.

A3.

在Kotlin，用來處理集合Collection的類型有List、Set、Map三種，而Map是使用Key-Value的結構來儲放多筆資料。Key會是唯一的，每一個key會對應到一個value，而value可以是重覆的

Exp1:

val numbersMap = mapOf("key1" to 1, "key2" to 2, "key3" to 3, "key4" to 1)

numbersMap.forEach { key, value ->

println("key:$key, value:$value")

}

//結果

**key:key1, value:1**

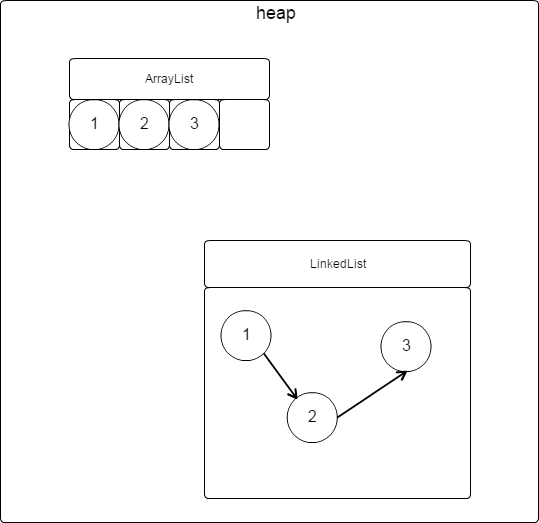
**key:key2, value:2**

**key:key3, value:3**

**key:key4, value:1**

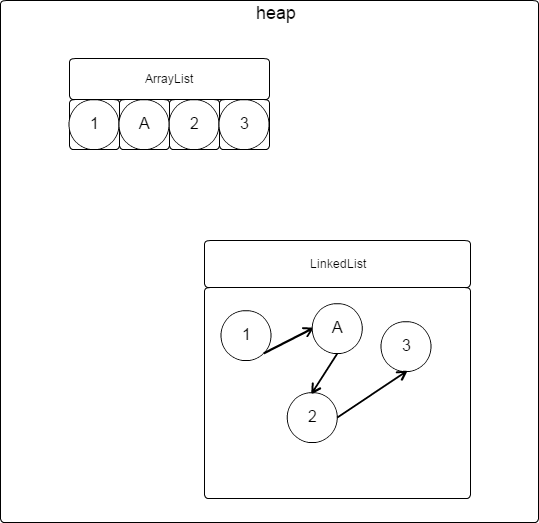
Q4. What’s the ​difference​ between ArrayList and LinkedList? Try to explain in Mandarin.

A4.

假設今天有三個Object分別利用ArrayList與LinkedList儲存，結果如下圖：

ArrayList是照順序儲存的，所以在取得特定的index元素時，非常方便快速，而LinkedList在取得特定的index元素，因為要一個一個去取，所以會比較慢。

但如果假設今天要插入第四個Object A，而且要接續在Object 1之後，會變成如下圖結果：



可以看到ArrayList的Object 2與Object 3都被移位了，而LinkedList則是修改Object 1的指向，再由Object A的指向Object 2，所以修改了二個指向，如果今天List內的數量很多，ArrayList會影響到被插入的index後的每一個，而LinkedList只會影響到二個。

總結：

|  |  |  |
| --- | --- | --- |
|  | ArrayList | LinkedList |
| 存放方式 | 陣列的方式 | 以節點連結方式 |
| 找特定index值 | 可以快速找到 | 因為需要重頭一個接續一個找，所以比較慢 |
| 插入中間值 | 影響中間index之後的每一個值，所以執行時間較久 | 只影響二個指向，所以執行時間較快 |
| 記憶體空間 | 長度是可變動的，並且會預留空間來存放資料 | 需要額外儲存上下節點的資訊，所以每個元素都會花費較多的記憶體 |

Q5. What are ​Value Type​ variables? List out the Value Type variables you know.

A5.

實值型別(Value Type)：如果資料型別是在其本身的記憶體中存放的是"實值"(Value)，資料型別就屬於"實值"型別。

實值型別共有8種，又可分為以下幾種類型：

一、整數類型：byte, short, int, long

二、浮點數類型：float, double

三、字元類型：char

四、布林類型：boolean

Q6. What’s the ​maximum​ value of ​Int​? If you want to represent integers larger than the limit,what should you do instead of using int variables?

Int值的範圍：最大值2147483647 (2 ^ 31 - 1) 到 最小值-2147483648(-2 ^ 31)

若使用超過或小於該資料表示的範圍，運算結果將會造成錯誤，也就是我們所說的資料溢位 (Overflow)，所以kotlin為所有數值類型都提供轉換方式，若是Int類型的值超過範圍，可用toLong()轉換為Long型態的整數值，Long範圍較大：為9223372036854775807 到

-9223372036854775808

Exp:

var bookPrice : Long = 2147483648.toLong()

println(bookPrice) **//2147483648**