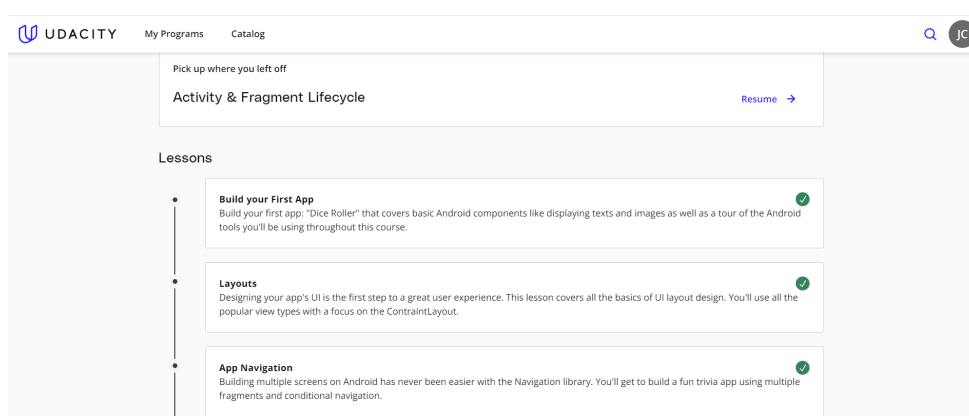


# homework2

## Screenshot



### 1

**inheritance** 可以將父類別中定義的 **attribute** 和 **method** 傳給子類別，使得子類別不需要重新定義也能使用這些性質。

### 2

If you want to split the string into a list of string with given size, you can use **.chunked()**

```
val dnaFragment = "ATTCGCGCCGCCAA"

val codons = dnaFragment.chunked(size = 3)

println(codons) // [ATT, CGC, GGC, CGC, CAA]
```

If you want to split the string to a list of lines delimited by any of the following character sequences: CRLF(\r\n), LF(\n) or CR(\r), use **.lines()**

```
var str = "12345\r\nnewewfweew\n12345"
str.lines().forEach { println(it) }
//12345
//ewewfweew
//12345
```

**.Split()** method can split the string into a list of string delimited by delimiters passed as parameter.

```
str = "hi,my,friend"
str.split(",").forEach { println(it) }
//hi my friend
```

### 3

We use Map, MutableMap, HashMap and LinkedHashMap to store key-value pairs in Kotlin.

```
val heights = mapOf("Jason" to 175, "Lulu" to 165)
for ((k, v) in heights) {
    println("$k: $v")
}
```

### 4

LinkedList是由一堆節點組成的，每個點除了儲存值以外，還會儲存下一個點的pointer，這使得LinkedList在儲存上會比較彈性，不需要一整塊連續的記憶體來存放。

在時間複雜度方面做比較，LinkedList只有在取值方面表現較差 $O(n)$ (假設這邊討論的都是DoublyLinkedList)，其他插入、刪除部分都是不劣於ArrayList的，但是ArrayList的使用場景遠多於LinkedList。

特別適和LinkedList的情形: 儲存並運算sparse

vector[1,0,0,0,0,0,0,1]，如果用一般的list，那會浪費許多空間放0，但對於LinkedList，(0,1) -> (8,1)就解決了

### 5

Value Type variables store the value while the reference Type variables store **the reference to a certain value**.

```
var a = 1
var b = a
a += 1
println(b) // b remains 1 while a becomes 2
```

As the code above, a stores the value 1, and when b is assigned to a, value 1 is replicated to b's value. That's the reason b remains unchanged after a's value is modified.

```
val arr = arrayOf(1, 2, 3, 4)
val arr2 = arr
arr[1] = 4
arr.forEach { println(it) } // 1, 4, 3, 4
arr2.forEach { println(it) } // 1, 4, 3, 4
```

However, if it is the case of Reference type variables, arr2 is assigned to the specific address the same as arr refers to. arr[1] = 4 changes the value keeping in that address. arr2 still refers to the address whose value is modified and

hence prints the same value as arr does.

Int, Long, Short, Byte, Float, Double, Boolean and Char are Value Type variables I know.

## 6

---

In Kotlin, Int is represented in 32bits. The first bit is sign bit, and the remaining 31bits can have  $2^{31}$  kinds of combinations. Hence, the range for integer value is from  $-2^{31}$  to  $2^{31} - 1$ .

First, I will use Long datatype which extends the limit to  $2^{63} - 1$ .

If the BigInt exceeds the limit of Long datatype, I will use an integer array to store the integer exceeding the limit. I think it is pretty easy to implement comparison, addition, subtraction ,multiplication and division on it.