Kotlin和java差別

# Chapter7

1. 隨機數產生：

Kotlin:

Kotlin 的 IntRange 有 random() 方法，可以直接從範圍中取得隨機數，，語法較簡潔。

val priceRange = IntRange(10, 100)

java:

Java 需要使用 Random 類別，nextInt(91) 會產生 0-90 的數字，且需要手動加上起始值 10，較為麻煩且複雜。

int price = 10 + random.nextInt(91);

1. 屬性存取

Kotlin:

直接使用屬性賦值，背後實際調用 setter 方法。

spinner.*adapter* = ArrayAdapter(this,  
 android.R.layout.*simple\_list\_item\_1*, count)

java:

Java 版本須直接呼叫 setter 方法，為更傳統的 Java 風格。

spinner.setAdapter(new ArrayAdapter<>(this,

android.R.layout.simple\_list\_item\_1, count));

1. 陣列使用方法

Kotlin:

使用 in 關鍵字，until 表示不包含最後一個值，較簡潔易讀。

for(index in 0 *until* array.length()) {

Java:

傳統的 for 迴圈，較冗長。

for (int index = 0; index < imageIds.length; index++) {

# Chapter8

1. 集合操作

Kotlin:

直接使用 add() 方法。

contacts.add(Contact(name, phone))  
myAdapter.notifyDataSetChanged()

java:

需要使用 new 關鍵字，需要分號結尾。

contacts.add(new Contact(name, phone));

myAdapter.notifyDataSetChanged();

## Chapter9-1

1. 隨機邏輯

Kotlin:

val sleepProbability = *arrayOf*(true, true, false)

if (sleepProbability.*random*())  
 Thread.sleep(300)

Java:

Boolean[] sleepProbability = {true, true, false};

if (Math.random() < 2.0 / 3.0)

Thread.sleep(300);

## Chapter9-2

1. 空值與輸入驗證

Kotlin:

使用when，且較簡潔。

when {  
 edHeight.*text*.*isEmpty*() -> showToast("請輸入身高")  
 edWeight.*text*.*isEmpty*() -> showToast("請輸入體重")  
 edAge.*text*.*isEmpty*() -> showToast("請輸入年齡")  
 else -> runThread() // 執行 runThread 方法  
}

Java:

使用多層 if-else，且看起來較為複雜。

if (edHeight.getText().toString().isEmpty()) {

showToast("請輸入身高");

} else if (edWeight.getText().toString().isEmpty()) {

showToast("請輸入體重");

} else if (edAge.getText().toString().isEmpty()) {

showToast("請輸入年齡");

} else {

runThread();}

1. 數學運算

Kotlin: (height / 100).pow(2)

Java: Math.pow(height / 100, 2)

Kotlin有內建的pow()函數

Kotlin:

使用Pair，較為簡潔。

val (standWeight, bodyFat) = if (btnBoy.*isChecked*) {  
 Pair((height - 80) \* 0.7, 1.39 \* bmi + 0.16 \* age - 19.34)  
} else {  
 Pair((height - 70) \* 0.6, 1.39 \* bmi + 0.16 \* age - 9)  
}

Java:

需要每個變數都分開宣告

final double standWeight, bodyFat;  
if (btnBoy.isChecked()) {  
 standWeight = (height - 80) \* 0.7;  
 bodyFat = 1.39 \* bmi + 0.16 \* age - 19.34;  
} else {  
 standWeight = (height - 70) \* 0.6;  
 bodyFat = 1.39 \* bmi + 0.16 \* age - 9;  
}

claud 問題

1. 瞭解這兩種程式的差別，還有Lambda的意思和其解釋



1. 詢問了Lambda，Java和Kotlin的差別



1. 想了解Pair功能



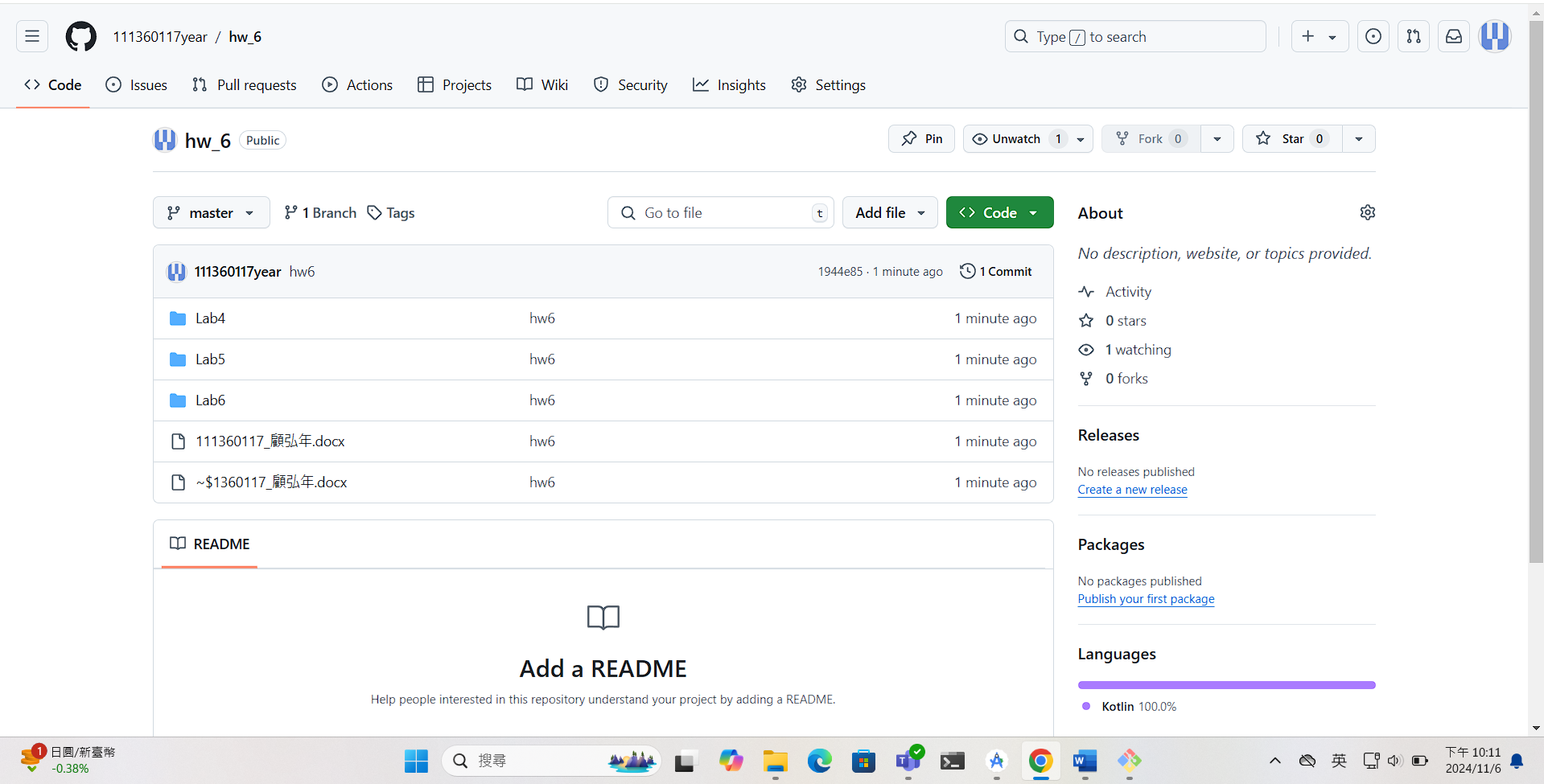
心得

這次練習也適用到許多沒用過的功能，像是Lab9的一些數學輸入和運算的函數，還有發現kotlin在表現這種需要整理較多資料的程式，都會有比較清晰簡潔的方法，還有java沒有的內建運算函數，相較之下我覺得kotlin讀起來是比較易讀一點，像是我有問Claud的第三點，就是Kotlin的功能之一，我覺得非常方便，能夠直接把兩個不同需要運算的變數在一行程式中完成，看起來也比較舒服一些。

Github

Github連結：<https://github.com/111360117year/hw_6.git>

Github截圖：



讀書會

組員：顧弘年、莊懷葰、林琮曜

時間：2024/11/06 23:15～ 2024/11/07 00:30

地點：Google meet

