Java 程式碼練習

心得

這次的作業老師要我們最佳化程式碼的可讀性，不要讓邏輯以 0、1、2 的方式出現。

在和同組的同學討論的過程中，我們第一個想到的是 C/C++ 中的 #define 指示器，能夠在不變更程式碼邏輯的情況下，使用比較有意義的單字來替代。

於是我們利用 ChatGPT，詢問他在這個方向上有沒有什麼在 Kotlin 中可以用的方法，他提示我們可以使用 enum class，並提供給我一些範例。

但為了自己搞懂這個功能在做什麼，我還是請他給我官方的文件，並且和我的同學在讀書會中自己閱讀討論了一下，後續再來嘗試改進程式碼的部分。

```
// 用 enum 取代 0/1/2
enum class Mora(val label: String) {
    SCISSOR("剪刀"),
    STONE("石頭"),
    PAPER("布");

    // 誰贏誰
    fun beats(other: Mora): Boolean = when (this) {
        SCISSOR -> other == PAPER
        STONE   -> other == SCISSOR
        PAPER   -> other == STONE
    }
}
```

```
            // Step9 判斷勝負
            when {
                myMora == targetMora -> {
                    tvWinner.text = "勝利者\n平手"
                    tvText.text = "平局，請再試一次！"
                }
                myMora.beats(targetMora) -> {
                    tvWinner.text = "勝利者\n$playerName"
                    tvText.text = "恭喜你獲勝了！！！"
                }
                else -> {
                    tvWinner.text = "勝利者\n電腦"
                    tvText.text = "可惜，電腦獲勝了！"
                }
            }
```

Kotlin 和 Java 在撰寫上有很多的不同，以我初學者的淺顯視角來看大概是 Kotlin 比較親切一點。兩者看起來都是物件導向的語言，我覺得 Lotlin 寫起來比較像 Python 一點；Java 寫起來則比較像是更往物件導向靠攏的 C++。

例如在變數的定義和元件的綁定上，寫法就有很多不一樣：

```
    private EditText ed_name;
    private TextView tv_text, tv_name, tv_winner, tv_mmora, tv_cmora;
    private RadioButton btn_scissor, btn_stone, btn_paper;
    private Button btn_mora;

    ...

        ed_name = findViewById(R.id.ed_name);
        tv_text = findViewById(R.id.tv_text);
        tv_name = findViewById(R.id.tv_name);
        tv_winner = findViewById(R.id.tv_winner);
        tv_mmora = findViewById(R.id.tv_mmora);
        tv_cmora = findViewById(R.id.tv_cmora);
        btn_scissor = findViewById(R.id.btn_scissor);
        btn_stone = findViewById(R.id.btn_stone);
        btn_paper = findViewById(R.id.btn_paper);
        btn_mora = findViewById(R.id.btn_mora);
```

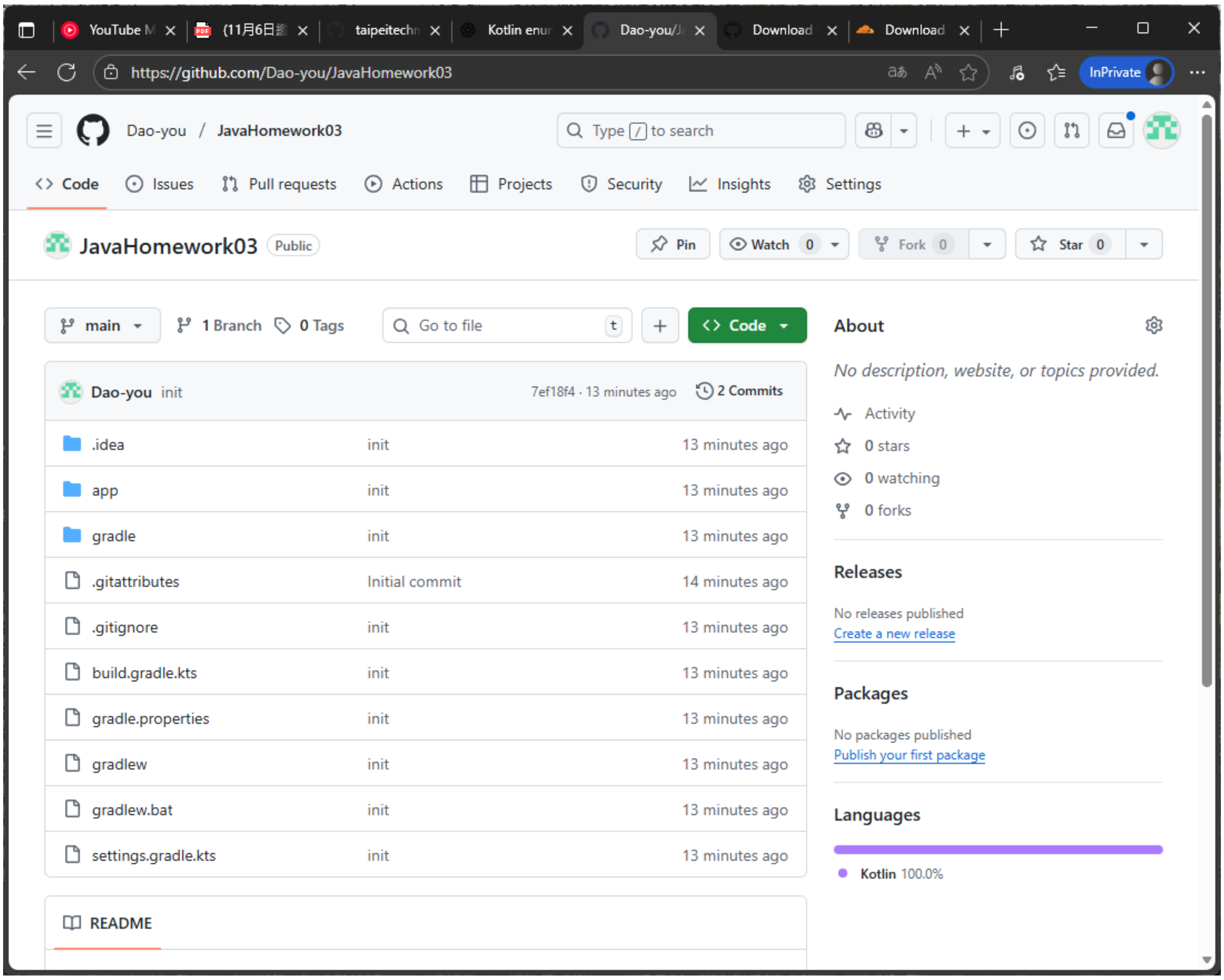
```
        val edName = findViewById<EditText>(R.id.edName)
        val tvText = findViewById<TextView>(R.id.tvText)
        val radioGroup = findViewById<RadioGroup>(R.id.radioGroup)
        val btnMora = findViewById<Button>(R.id.btnMora)
        val tvName = findViewById<TextView>(R.id.tvName)
        val tvWinner = findViewById<TextView>(R.id.tvWinner)
        val tvMyMora = findViewById<TextView>(R.id.tvMyMora)
        val tvTargetMora = findViewById<TextView>(R.id.tvTargetMora)
```

Github

網址：

https://github.com/Dao-you/JavaHomework03

截圖：



https://github.com/Dao-you/JavaHomework03

Dao-you / JavaHomework03

JavaHomework03 Public

main · 1 Branch · 0 Tags

Dao-you init · 7ef18f4 · 13 minutes ago · 2 Commits

.idea	init	13 minutes ago
app	init	13 minutes ago
gradle	init	13 minutes ago
.gitattributes	Initial commit	14 minutes ago
.gitignore	init	13 minutes ago
build.gradle.kts	init	13 minutes ago
gradle.properties	init	13 minutes ago
gradlew	init	13 minutes ago
gradlew.bat	init	13 minutes ago
settings.gradle.kts	init	13 minutes ago

README

About

No description, website, or topics provided.

Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

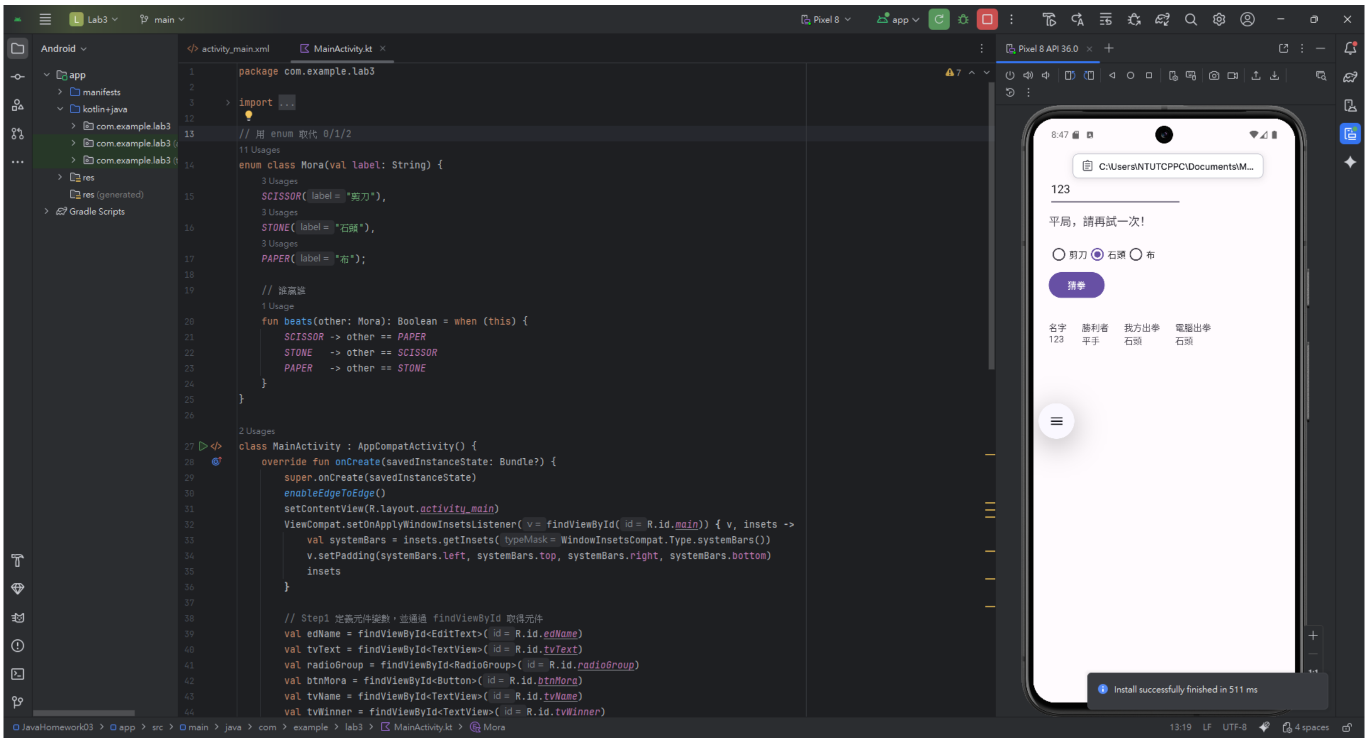
Packages

No packages published
Publish your first package

Languages

Kotlin 100.0%

程式碼執行



MainActivity.kt

```kotlin
package com.example.lab3

import ...

// 用 enum 取代 0/1/2
enum class Mora(val label: String) {
    SCISSOR("剪刀"),
    STONE("石頭"),
    PAPER("布");

    // 誰贏誰
    fun beats(other: Mora): Boolean = when (this) {
        SCISSOR -> other == PAPER
        STONE   -> other == SCISSOR
        PAPER   -> other == STONE
    }
}

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        // Step1 定義元件變數，並通過 findViewById 取得元件
        val edName = findViewById<EditText>(R.id.edName)
        val tvText = findViewById<TextView>(R.id.tvText)
        val radioGroup = findViewById<RadioGroup>(R.id.radioGroup)
        val btnMora = findViewById<Button>(R.id.btnMora)
        val tvName = findViewById<TextView>(R.id.tvName)
        val tvWinner = findViewById<TextView>(R.id.tvWinner)
```

Pixel 8 API 36.0

123

平局，請再試一次!

○ 剪刀 [✓ 石頭] ○ 布

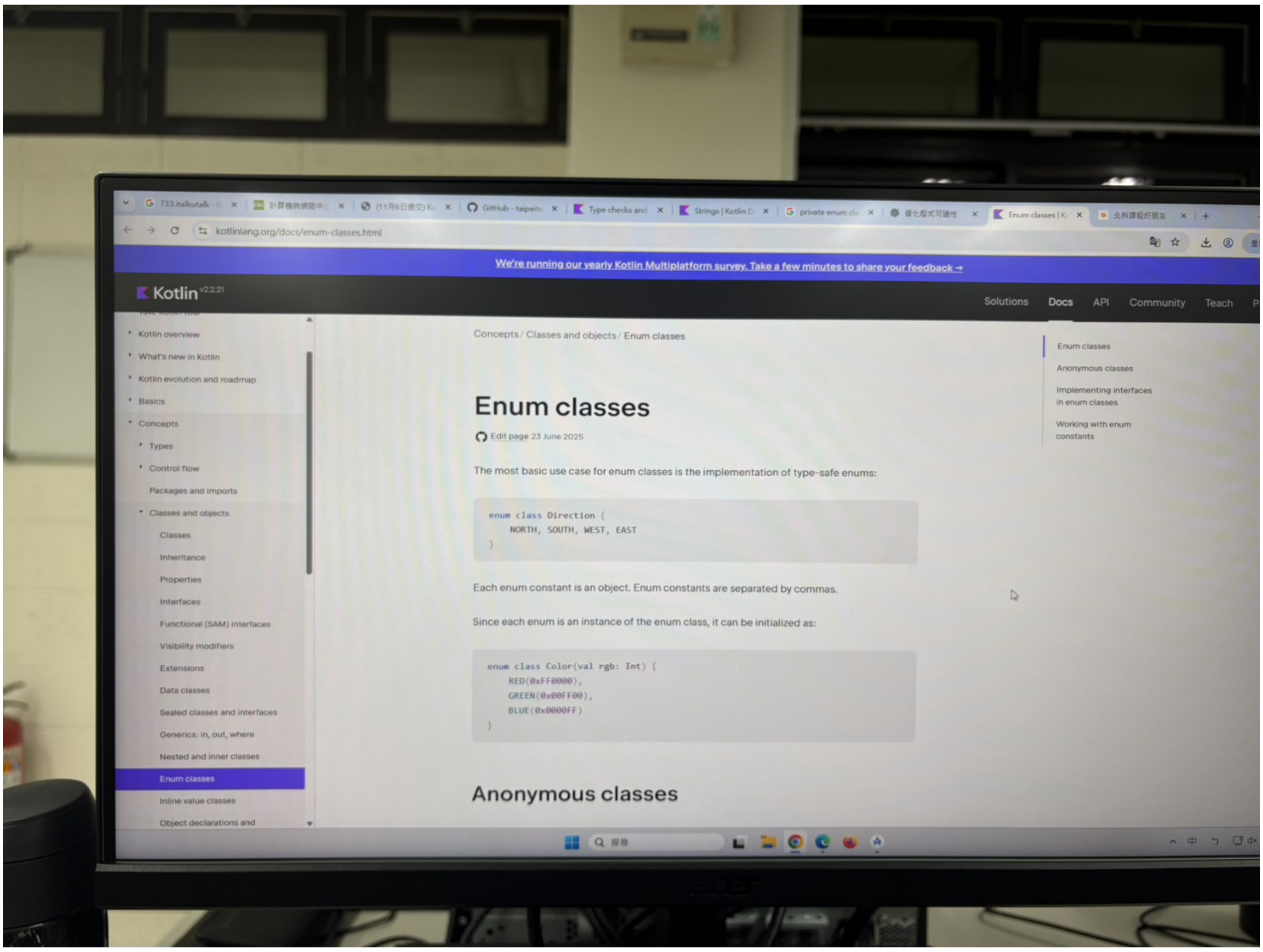
猜拳

名字	勝利者	我方出拳	電腦出拳
123	平手	石頭	石頭

Install successfully finished in 511 ms

讀書會

- 組員：方宇澤
- 討論時間：2025/10/31 15:00~16:00
- 地點：電腦教室(做完實驗之後討論)
- 討論主題：程式碼可讀性最佳化、Kotlin 基本、enum class 的使用



Concepts / Classes and objects / Enum classes

Enum classes

Edit page 23 June 2025

The most basic use case for enum classes is the implementation of type-safe enums:

```
enum class Direction {
    NORTH, SOUTH, WEST, EAST
}
```

Each enum constant is an object. Enum constants are separated by commas.

Since each enum is an instance of the enum class, it can be initialized as:

```
enum class Color(val rgb: Int) {
    RED(0xFF0000),
    GREEN(0x00FF00),
    BLUE(0x0000FF)
}
```

Anonymous classes