

讀書會

- 組員：龍品勳、張耀文、時顯文
- 討論時間：2024/12/18 10:00 AM
- 地點：圖書館

照片：



心得報告

- 心得：執行緒很有趣。
- ChatGPT：

1. 請使用 `java.applet` 寫出範例程式碼。

以下是一個使用 `java.applet.Applet` 的範例程式碼，這段程式碼簡單地繪製了一個圖和顯示一段文字。請注意，`java.applet` 是一個過時的技術，在較新的 Java 版本中可能不被支援，因此建議在支援的環境中執行此程式碼。

Applet Example

在交談中回答 >

編譯與執行

- 將上述程式碼儲存為 `SimpleApplet.java`。
- 在命令提示字元中執行以下命令編譯程式：

```
bash
javac SimpleApplet.java
```

- 建立一個 HTML 檔案 (例如 `SimpleApplet.html`)，內容如下：

```
html
<html>
<body>
<applet code="SimpleApplet.class" width="300" height="300"></applet>
</body>
</html>
```

- 使用支援的 Java 瀏覽器或 Applet Viewer 開啟 HTML 檔案：

```
bash
appletviewer SimpleApplet.html
```

注意事項

- 從 Java 9 開始，`java.applet` API 已被標記為過時，並在 Java 17 中被完全移除。建議考慮替代技術，如 `JavaFX` 或現代 GUI 框架。

Applet Example

```
1 import java.applet.Applet;
2 import java.awt.*;
3
4 // 繼承 Applet 類別
5 public class SimpleApplet extends Applet {
6
7     // 初始化方法，在 Applet 加載時調用
8     @Override
9     public void init() {
10         setBackground(Color.WHITE); // 設定背景顏色
11     }
12
13     // 繪製方法，繪製內容會顯示在 Applet 中
14     @Override
15     public void paint(Graphics g) {
16         g.setColor(Color.BLUE); // 將繪圖筆顏色設為藍色
17         g.drawString("Hello, Applet!", 50, 50); // 繪製文字
18
19         g.setColor(Color.RED); // 設定繪圖筆顏色為紅色
20         g.fillOval(50, 70, 100, 100); // 繪製一個實心圓形
21     }
22 }
23
```

2. 請使用 `java.io` 寫出範例程式碼。



以下是一個使用 Java 的 `java.io` 包來進行檔案讀取和寫入的範例程式碼。這個程式會從一個檔案讀取文字內容，然後將內容寫入另一個檔案：

範例程式碼

java

複製程式碼

```
import java.io.*;

public class FileReadWriteExample {
    public static void main(String[] args) {
        // 定義檔案路徑
        String inputFile = "input.txt";
        String outputFile = "output.txt";

        // 使用 try-with-resources 自動管理資源
        try (
            BufferedReader reader = new BufferedReader(new FileReader(inputFile));
            BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile))
        ) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println("讀取內容: " + line); // 印出讀取內容
                writer.write(line); // 寫入到輸出檔案
                writer.newLine(); // 新增換行
            }
            System.out.println("檔案寫入完成: " + outputFile);
        } catch (FileNotFoundException e) {
            System.err.println("檔案未找到: " + inputFile);
        } catch (IOException e) {
            System.err.println("讀取或寫入時發生錯誤: " + e.getMessage());
        }
    }
}
```



如何使用

1. 創建一個名為 `input.txt` 的檔案，並在裡面寫入一些文字內容。
2. 執行程式後，檔案內容會被讀取並寫入到名為 `output.txt` 的新檔案中。
3. 若檔案不存在或讀寫過程中發生錯誤，會顯示相應的錯誤訊息。

程式解說

- `FileReader` 和 `FileWriter`：用於檔案的基本讀取與寫入操作。
- `BufferedReader` 和 `BufferedWriter`：提供緩衝功能，提升讀寫效能。
- `try-with-resources`：確保資源在使用完畢後會自動關閉，不需要手動呼叫 `close()` 方法。
- 例外處理：捕捉 `FileNotFoundException` 和 `IOException` 以處理可能的錯誤。

如果需要更多的範例或進階功能，請告訴我！

3. 請使用 `java.util` 寫出範例程式碼。



以下是一個簡單的範例程式碼，使用 Java 中的 `java.util` 包來展示以下功能：

- 使用 `ArrayList` 來儲存資料
- 使用 `Collections` 進行排序
- 使用 `HashMap` 來儲存鍵值對
- 使用 `Scanner` 從使用者輸入

Java 程式碼範例

java

複製程式碼

```
import java.util.*;

public class JavaUtilExample {
    public static void main(String[] args) {
        // 使用 ArrayList 儲存資料
        List<String> names = new ArrayList<>();
        names.add("Alice");
        names.add("Bob");
        names.add("Charlie");

        // 輸出原始資料
        System.out.println("原始名單: " + names);

        // 使用 Collections 排序
        Collections.sort(names);
        System.out.println("排序後的名單: " + names);

        // 使用 HashMap 儲存鍵值對
        Map<String, Integer> ageMap = new HashMap<>();
        ageMap.put("Alice", 25);
        ageMap.put("Bob", 30);
        ageMap.put("Charlie", 20);
    }
}
```



```

System.out.println("\n名單與年齡:");
for (String name : names) {
    System.out.println(name + " 的年齡是 " + ageMap.get(name));
}

// 使用 Scanner 讀取使用者輸入
Scanner scanner = new Scanner(System.in);
System.out.print("\n請輸入一個名字來查詢年齡: ");
String inputName = scanner.nextLine();

// 查詢並顯示結果
if (ageMap.containsKey(inputName)) {
    System.out.println(inputName + " 的年齡是 " + ageMap.get(inputName));
} else {
    System.out.println("找不到 " + inputName + " 的資料!");
}

scanner.close();
}
}

```

複製程式碼

執行結果範例

假設使用者輸入 `Alice`：

makefile

複製程式碼

原始名單: [Alice, Bob, Charlie]
 排序後的名單: [Alice, Bob, Charlie]

名單與年齡:
 Alice 的年齡是 25
 Bob 的年齡是 30
 Charlie 的年齡是 20

請輸入一個名字來查詢年齡: Alice
 Alice 的年齡是 25



功能說明

1. **ArrayList**：動態陣列，適合儲存列表資料。
2. **Collections.sort**：方便進行排序操作。
3. **HashMap**：儲存鍵值對，適合快速查詢。
4. **Scanner**：從使用者輸入取得資料。

如果有其他需求，歡迎告訴我！

- GitHub 程式連結：<https://github.com/WoemTheCat/CH5>
- GitHub 的截圖：

