**Lab#12 – Analyze the quality of your software by using software metrics**

# **วัตถุประสงค์การเรียนรู้**

1. ผู้เรียนสามารถอธิบายความหมายของ Software metrics ที่วัดออกมาได้อย่างถูกต้อง
2. ผู้เรียนสามารถใช้ Static code analysis tools ในการวิเคราะห์ปัญหาคุณภาพของซอฟต์แวร์ได้
3. ผู้เรียนสามารถอธิบายความแตกต่างระหว่าง Static testing และ Dynamic testing ได้อย่างถูกต้อง

**กิจกรรมที่ 12.1: การวิเคราะห์ปัญหาในโค้ดด้วย PMD**

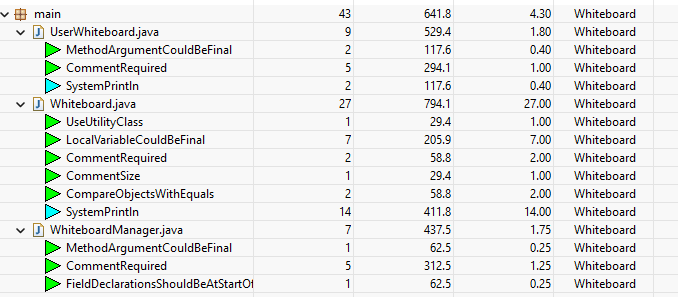
1. ติดตั้ง PMD plugin ลงใน Eclipse ให้เรียบร้อย
2. สร้าง local repository ของ git และตั้งค่าให้เรียบร้อย Workspace ให้เรียบร้อย
3. Clone โค้ดตั้งต้นจาก

<https://github.com/ChitsuthaCSKKU/SQA/tree/2025/Assignment/Lab12_SoftwareMetrics>

1. ศึกษาการทำงานและโครงสร้างของ Source code ที่กำหนดให้
2. วิเคราะห์ปัญหาใน Source code โดยใช้ PMD
3. Capture หน้าจอ หรือ Export รายงานผลการทดสอบ แล้วตอบคำถามต่อไปนี้

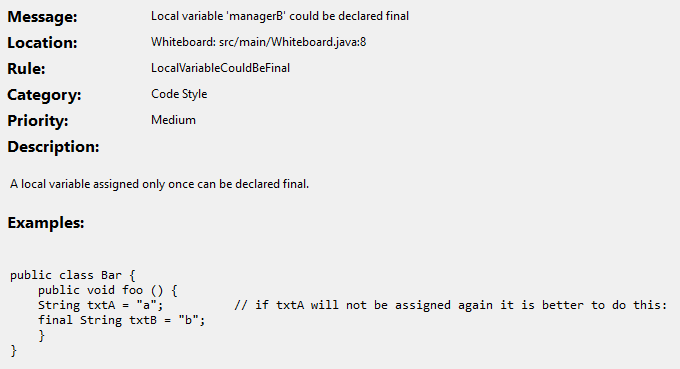
**Q1: PMD พบ Error และ Warning ใน Whitboard.java กี่ตัว (Capture หน้าจอประกอบคำตอบ)**

ตอบ 27 ตัว



**Q2: จงอธิบายเกี่ยวกับปัญหา “Local variable 'managerB' could be declared final” ว่าเป็นปัญหาอะไร และมีแนวทางในการแก้ไขอย่างไร**

ตอบ ปัญหาจากการที่ตัวแปรนี้ใช้เก็บค่าค่าเดียว โดยที่จะไม่เปลี่ยนค่าอีก ควร assign เป็น final ไว้เลย



**Q3: PMD ใช้ Metrics อะไรบ้างในการระบุ God class และแต่ละตัวมีความหมายว่าอะไรบ้าง**

ตอบ 1. WMC (Weighted Methods per Class): วัดจำนวนและความซับซ้อนของ method ภายใน class นั้น เช่น ถ้าคลาสนี้มี method 30 ตัวถือว่าเป็น God class

2. ATFD (Access to Foreign Data): วัดจำนวนครั้งที่ class นี้เข้าถึง data (field/method) จาก class อื่น เช่น class A เรียกใช้ getter/setter หรือเข้าถึง field จาก class B, C, D เยอะ คะแนน ATFD จะสูง

3. TCC (Tight Class Cohesion): วัดระดับความเชื่อมโยง (cohesion) ระหว่าง methods ภายใน class ถ้ายิ่งต่ำแสดงว่า methods ทำงานแยกส่วนกัน หมายถึง class นี้อาจทำหลายหน้าที่เกินไป

**กิจกรรมที่ 12.2: การวิเคราะห์ปัญหาในโค้ดด้วย Checkstyle**

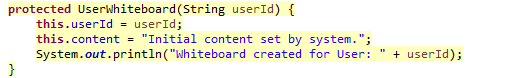
1. ติดตั้ง checkstyle plugin ลงใน Eclipse ให้เรียบร้อย
2. สร้าง local repository ของ git และตั้งค่าให้เรียบร้อย Workspace ให้เรียบร้อย
3. Clone โค้ดตั้งต้นจาก

<https://github.com/ChitsuthaCSKKU/SQA/tree/2025/Assignment/Lab12_SoftwareMetrics>

1. ศึกษาการทำงานและโครงสร้างของ Source code ที่กำหนดให้
2. วิเคราะห์ปัญหาใน Source code โดยใช้ checkstyle
3. Capture หน้าจอ หรือ Export รายงานผลการทดสอบ แล้วตอบคำถามต่อไปนี้

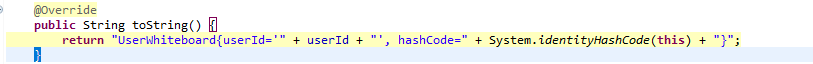
**Q4: จงระบุปัญหาการเขียน Code ที่ไม่ตรงกับ Coding standard ของ Java ที่ Checkstyle ระบุมา 3 รายการ พร้อมคำอธิบายว่า Code ส่วนนี้ไม่สอดคล้องกับมาตรฐานอย่างไร (Capture หน้าจอประกอบด้วย)**

ตอบ 1. Missing a Javadoc comment.



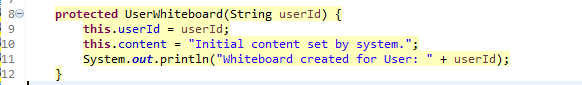
ฟังก์ชันนี้ไม่มี comment บอกว่าใช้ทำอะไร

2. Line is longer than 100 characters (found 105).



บรรทัดนี้มีความยาวโค้ดมากเกินกว่า 100 ตัวอักษร

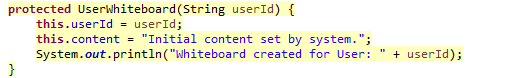
3. 'method def' child has incorrect indentation level 8, expected level should be 4.



บรรทัดที่ 9, 10 และ 11 มีการเยื้องมากเกินไป ตอนนี้มี level 8 แต่ที่จริงควรมี level 4

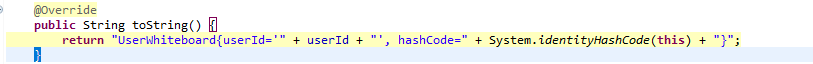
**Q5: จาก Q4 หากต้องการแก้ไข Source code เพื่อให้ผ่านมาตรฐานการเขียนโค้ดและการตรวจสอบโดย Checkstyle จะต้องแก้ไข Code ในบรรทัดนั้น ๆ อย่างไร**

ตอบ 1. Missing a Javadoc comment.



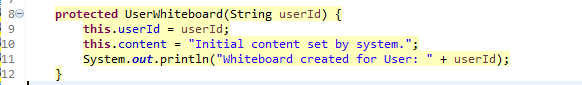
ให้เพิ่ม comment บอกว่าฟังก์ชันนี้ใช้ทำอะไร

2. Line is longer than 100 characters (found 105).



ลดความยาวของข้อความลงมา หรือกด enter ตรงบรรทัดนี้ให้ขึ้นบรรทัดใหม่

3. 'method def' child has incorrect indentation level 8, expected level should be 4.



ลดระยะการเยื้องของบรรทัดที่ 9, 10 และ 11 ลงให้เหลือเพียง 4 spaces และลดบรรทัดที่ 8 และ 12 ตามลงมาด้วย