

# Lab 11: Word Count

## วัตถุประสงค์

- เพื่อให้นิสิตสามารถแยกข้อมูลในสายอักขระเป็นส่วน ๆ ได้
- เพื่อให้นิสิตสามารถใช้งาน JUnit ได้

## ปัญหาที่ 1: การนับจำนวนคำในสายอักขระ (String)

ในปัญหาข้อนี้ นิสิตจะได้ทบทวนการใช้ JUnit ในการตรวจสอบการทำงานของโปรแกรม โดยเราจะพิจารณาปัญหาการนับจำนวนคำในสายอักขระ นิสิตจะต้องทำการสร้างคลาสที่ทำหน้าที่รับข้อมูลประเภทสายอักขระ และทำการคำนวณจำนวนคำและ ความยาวเฉลี่ยของคำ โดยที่คลาสดังกล่าวจะต้องมีเมทอดตามที่กำหนดให้

คำในสายอักขระนั้นคือกลุ่มตัวอักษร รวมถึงตัวเลข และเครื่องหมายใด ๆ ที่เรียงต่อกันโดยไม่มีช่องว่างคั่น ส่วนช่องว่างนั้นไม่นับเป็นคำ ตัวอย่างเช่น “this is a string.” (ให้สังเกตว่ามีส่วนที่เป็นช่องว่างติดกัน) เป็นสายอักขระที่มีคำทั้งหมด 4 คำ คือ “this”, “is”, “a”, “string.” ความยาวของแต่ละคำนั้นคือจำนวนของตัวอักษร, ตัวเลข และเครื่องหมายใด ๆ ในคำนั้น ดังเช่นตัวอย่างที่แล้ว ความยาวของแต่ละคำ คือ 4, 2, 1 และ 7 และความยาวเฉลี่ยของสายอักขระนั้นคือ  $(4+2+1+7) / 4 = 3.75$

ในการปฏิบัติการครั้งนี้ ให้นิสิตทำการสร้าง Project ชื่อ Lab11 และให้นำเข้าแฟ้มชื่อ WordCount.jar ซึ่งจะมีแฟ้มชื่อ WordCount.java อยู่ภายใน ในแฟ้มดังกล่าวจะมีคลาส WordCount พร้อมทั้งโครงของเมทอดจำนวนสี่ เมทอด ให้นิสิตทำการแก้ไขเมทอดทั้งสี่ ตามหน้าที่การทำงานดังต่อไปนี้ โดยห้ามแก้ไขโครงของเมทอดดังกล่าว

<code>void setText(String text)</code>	เป็นเมทอดที่ทำหน้าที่รับสายอักขระเข้ามา
<code>int getWordCount()</code>	เมทอดนี้จะ <b>return</b> จำนวนคำในสายอักขระที่รับเข้ามาผ่านทางเมทอด <b>setText</b>
<code>double getAvgWordLength()</code>	เมทอดนี้จะ <b>return</b> ความยาวเฉลี่ยของคำในสายอักขระที่รับเข้ามาผ่านทางเมทอด <b>setText</b> ถ้าสายอักขระที่รับเข้ามานั้นไม่มีคำ ให้ <b>return</b> ค่า -1
<code>String getAvgWordLengthAsString()</code>	เมทอดนี้จะ <b>return</b> ความยาวเฉลี่ยของคำในสายอักขระที่รับเข้ามาผ่านทางเมทอด <b>setText</b> โดยที่ ความยาวเฉลี่ยที่ <b>return</b> กลับไปนั้นจะถูกส่งกลับไปในรูปแบบสายอักขระโดยมีทศนิยมจำนวนสองตำแหน่ง ตัวอย่างเช่น ถ้าเราเรียก <b>setText("this test")</b> เมทอดนี้จะ <b>return</b> ค่า “4.00” เป็นต้น และถ้าสายอักขระที่รับเข้ามานั้นไม่มีคำอยู่เลย ให้ <b>return</b> ค่า “-1.00”

\*\*\* งาน 1: สร้าง Project และนำเข้าแฟ้มต่าง ๆ \*\*\*

ส่วนของโปรแกรมต่อไปนี้จะแสดงถึงการใช้งานคลาส WordCount

```
WordCount wc = new WordCount();

wc.setText("this test");
System.out.println("word count = "+wc.getWordCount());
System.out.println("average word length = "+wc.getAvgWordLength());
System.out.println("formatted average word length =
    "+wc.getAvgWordLengthAsString());
```

สำหรับคลาส WordCount ที่ถูกต้อง ผลการทำงานของส่วนของโปรแกรมข้างต้นจะเป็นดังนี้

```
word count = 2
average word length = 4
formatted average word length = 4.00
```

**\*\*\* งาน 2: แก้ไขเมทอดตามที่ได้กำหนดไว้ \*\*\***

เมื่อนิสิตทำการแก้ไขเมทอดดังกล่าวเสร็จสิ้นแล้ว ให้นิสิตทดสอบการทำงานของแต่ละเมทอดโดยใช้ JUnit ซึ่งอยู่ในแฟ้ม TestWordCount.java ให้นิสิตทำการตั้งค่าให้ Eclipse นั้นสามารถใช้งาน JUnit version 4 ได้ และทำการทดสอบโปรแกรมที่ได้เขียนขึ้นมาพร้อมทั้งแก้ไขจนกระทั่งโปรแกรมของนิสิตสามารถผ่าน Test Case ทั้งหมดใน TestWordCount.java

**\*\*\* งาน 3: เรียกใช้งาน JUnit พร้อมทั้งแก้ไขโปรแกรมให้ผ่านการทดสอบของ JUnit \*\*\***

## คำแนะนำ

- ในการปฏิบัติภารกิจครั้งนี้ นิสิตจะต้อง “แยก” สายอักขระออกมาเป็นส่วน ๆ โดยแต่ละส่วนจะเป็นคำหนึ่งคำ การแยกส่วนดังกล่าวนั้นสามารถทำได้โดยใช้คลาส Scanner ดังนี้

```
Scanner scn = new Scanner("this is a string");
while (scn.hasNext()) {
    String word = scn.next();
    System.out.println(word);
}
```

ส่วนของโปรแกรมนี้จะใช้ scanner มาทำการตัดสายอักขระ “this is a string” ออกเป็นส่วน ๆ โดยที่แต่ละส่วนนั้นจะอยู่ในตัวแปรชื่อ word

- จากตัวอย่างข้างต้น เราจะเห็นได้ว่า คลาส Scanner นั้นสามารถรับข้อมูลเข้าได้หลายประเภท ในครั้งก่อน ๆ เราได้ใช้ Scanner นั้นรับข้อมูลเข้ามาจากแฟ้ม ส่วนครั้งนี้เรารับข้อมูลเข้ามาจากสายอักขระ นิสิตควรจะศึกษาการใช้งานของ Scanner ให้คล่องแคล่ว

2110271 **Programming Tool Labs**

ชื่อ-สกุล: \_\_\_\_\_ ID: \_\_\_\_\_ วัน: \_\_\_\_\_ เวลา \_\_\_\_\_ .

---

**LAB 11: Array Search & Modify**

---

<b>Task</b>	<b>Description</b>	<b>Grader's Signature</b>
1	Create and import .jar file	
2	Complete all methods	
3	Use and pass all test cases	