

CHAPTER 05

การดำเนินการกับสตริง (String Operations)

1. ลักษณะทั่วไปของสตริง (Characteristic of String)

1. นิยามและข้อมูลเบื้องต้นของสตริง

- 1) สตริง (String) หรือ สายอักขระ คือ อักขระ (Character) ตั้งแต่ศูนย์ตัวขึ้นไปมาเรียงต่อกันอยู่ภายใต้เครื่องหมาย Double Quote เช่น "Hello", "F", "@123", "คำถึง หนองกระโทก", " ", " " เป็นต้น
- 2) เรียก " " (เครื่องหมาย Double Quote ติดกัน) ว่า สตริงว่าง (Empty String / Null String) ซึ่งเป็นสตริงที่มีความยาวหรือจำนวนอักขระเป็นศูนย์

- 3) การประกาศตัวแปรประเภทสตริงทำได้ดังนี้

```
String <ชื่อสตริง> = "<ค่าสตริง>";
```

เช่น String s = "Hello Engineering";

คำว่า String จะต้องเขียนขึ้นต้นด้วย
อักษร S แบบพิมพ์ใหญ่เสมอ

ค่าของสตริงต้องล้อมด้วย "..." เสมอ

- 4) การนำเข้าสู่สตริงจากแป้นพิมพ์ จะใช้เมทอดของคลาส Scanner ซึ่งได้กล่าวไปแล้วในบทที่ 2 โดยประกอบไปด้วยเมทอดดังนี้

- nextLine() เป็นการรับสตริงทีละบรรทัด
- next() เป็นการรับสตริงทีละช่วงหรือทีละคำ (รับสตริงช่วงแรกหรือคำแรกจากซ้ายมือ)

- 5) สตริงมีตัวดำเนินการพื้นฐานดังต่อไปนี้

รูปแบบ	ข้อมูลที่ 1	กลุ่มของตัวดำเนินการ	ข้อมูลที่ 2	ประเภทข้อมูลสุดท้าย
1.	String	=	String	String
2.	String	+	ทุกประเภท	String (การต่อสตริง)
3.	ทุกประเภท	+	String	String (การต่อสตริง)
4.	String	== !=	String	boolean

2. โครงสร้างของสตริง

- 1) สตริงเป็นคลาสมาตรฐานในภาษาจาวา โดยโครงสร้างภายในเขียนด้วยอาร์เรย์ของตัวอักขระ (Array of Character) ดังตัวอย่าง

- สตริงในรูปทั่วไป String s = "Java";
- สตริงในรูปอาร์เรย์ char s[] = {'J', 'a', 'v', 'a'};

หรืออาจจะเขียนสตริงในรูปของอ็อบเจกต์ดังนี้
String s = new String("Java");

- 2) ตำแหน่ง (Index) ของสตริงจะเริ่มนับที่ 0 แต่ความยาว (Length) ของสตริงจะเริ่มนับที่ 1 ดังตัวอย่าง

```
String s = "JAVA CHULA.";
```

0	1	2	3	4	5	6	7	8	9	10
J	A	V	A		C	H	U	L	A	.

หมายเลขตำแหน่ง (Index)

สตริงยาว 11 อักขระ แต่ตำแหน่งสูงสุดคือ 10

- 3) สตริงมีเมทอดให้เรียกใช้งานมากมายดังรายละเอียดในหัวข้อถัดไป

2. การดำเนินการที่เกี่ยวกับสตริง (Operations of String)

1. วิธีเรียกใช้เมทอดจากคลาส String

เนื่องจากสตริงเป็นตัวแปรที่ถูกสร้างขึ้นจากคลาส String ดังนั้นการดำเนินการส่วนใหญ่ของสตริงจะต้องกระทำภายใต้เมทอดของสตริงหรือเมทอดที่เกี่ยวข้องกับสตริง โดยมีวิธีการเรียกใช้เมทอดดังนี้

`<ชื่อสตริง> . <ชื่อเมทอด> ([พารามิเตอร์])`

เช่น `s.length()`, `n.charAt(4)` เป็นต้น

2. เมทอดจากคลาส String

1) เมทอดหาความยาวของสตริง (String Length)

`<สตริง> .length()`

จะคืนค่าจำนวนเต็ม ซึ่งเป็นความยาวของสตริง

ตัวอย่างเช่น

```
String name = "Suda Rukchula";
int len = name.length(); //len = 13
```

2) เมทอดตัดช่องว่างทางซ้ายและขวาของสตริงออกไป (String Trim)

`<สตริง> .trim()`

จะคืนค่าสตริงที่เหมือนกับสตริงเดิมแต่ตัดช่องว่าง (Space) ทางซ้ายและขวาของสตริงออกไป

ตัวอย่างเช่น

```
String m = "    Hello JAVA    ";
String n = m.trim(); //n = "Hello JAVA"
```

3) เมทอดแปลงสตริงให้เป็นอักษรตัวพิมพ์ใหญ่ (String to Upper Case)

`<สตริง> .toUpperCase()`

จะคืนค่าสตริงที่เหมือนกับสตริงเดิมแต่ตัวอักษรอังกฤษทุกตัวจะเปลี่ยนเป็นตัวพิมพ์ใหญ่

ตัวอย่างเช่น

```
String m = "Hello! JaVa 2010";
m = m.toUpperCase(); //m = "HELLO! JAVA 2010"
```

4) เมทอดแปลงสตริงให้เป็นอักษรตัวพิมพ์เล็ก (String to Lower Case)

`<สตริง> .toLowerCase()`

คืนค่าสตริงที่เหมือนกับสตริงเดิมแต่ตัวอักษรอังกฤษทุกตัวจะเปลี่ยนเป็นตัวพิมพ์เล็ก

ตัวอย่างเช่น

```
String m = "$Hello JAVA #2553";
m = m.toLowerCase(); //m = $hello java #2553
```

5) เมทอดตัดสตริงย่อย (Substring)

`<สตริง> .substring(<เริ่มต้น> , <สิ้นสุด>)`

เป็นตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดของสตริงย่อย

คืนค่าสตริงที่เป็นสตริงย่อยตั้งแต่ตำแหน่งเริ่มต้นจนถึงก่อนตำแหน่งสิ้นสุด

ตัวอย่างเช่น

```
String name = "Chulalongkorn University";
String s1 = name.substring(9, 16); //s1 = "korn Un"
String s2 = name.substring(0, 5) + "Thailand"; //s2 = "ChulaThailand"
```

`<สตริง>.substring(<ตำแหน่ง>)`

คืนค่าสตริงที่เป็นสตริงย่อยตั้งแต่ตำแหน่งที่ระบุ จนถึงตำแหน่งสุดท้าย

ตัวอย่างเช่น

```
String name = "Chulalongkorn University";
String s1 = name.substring(14); //s1 = "University"
String s2 = name.substring(20); //s2 = "sity"
```

6) เมื่อกู้หาตัวอักขระ ณ ตำแหน่งที่ระบุในสตริง (Character At)

`<สตริง>.charAt(<ตำแหน่ง>)`

คืนค่าอักขระในตำแหน่งที่ระบุ

ตัวอย่างเช่น

```
String m = "Hello JAVA";
char ch1 = m.charAt(0); //ch1 = 'H'
char ch2 = m.charAt(6); //ch2 = 'J'
```

7) เมื่อกู้หาตำแหน่งที่พบเป็นครั้งแรกของสตริงย่อยในสตริงหลัก (Index Of)

`<สตริง>.indexOf(<สตริง>)`

คืนค่าจำนวนเต็มที่เป็นตำแหน่งของสตริงย่อยในสตริงหลักที่พบเป็นครั้งแรก ถ้าไม่พบจะคืนค่า -1

ตัวอย่างเช่น

```
String m = "Hello JAVA";
int i = m.indexOf("l"); //i = 2
int j = m.indexOf("lo"); //j = 3
int k = m.indexOf("F"); //k = -1
```

8) เมื่อกู้หาตำแหน่งที่พบเป็นครั้งสุดท้ายของสตริงย่อยในสตริงหลัก (Last Index Of)

`<สตริง>.lastIndexOf(<สตริง>)`

คืนค่าจำนวนเต็มที่เป็นตำแหน่งของสตริงย่อยในสตริงหลักที่พบเป็นครั้งสุดท้าย ถ้าไม่พบจะคืนค่า -1

ตัวอย่างเช่น

```
String m = "Hello JAVA";
int i = m.lastIndexOf("A"); //i = 9
int j = m.lastIndexOf("AV"); //j = 7
int k = m.lastIndexOf("b"); //k = -1
```

เริ่มนับที่ตำแหน่งที่ 0 (ซ้ายไปขวา)

9) เมื่อกู้หาเปรียบเทียบการเท่ากันทุกประการของสตริง

`<สตริง>.equals(<สตริง>)`

คืนค่าความจริงที่เป็น true ถ้าสตริงเท่ากันทุกประการ หรือ false ถ้าสตริงไม่เท่ากันทุกประการ

ตัวอย่างเช่น

```
String s1 = "E-mail";
String s2 = "e-mail";
boolean flag = s1.equals(s2); //flag = false
```

เปรียบเทียบสตริง s1 กับสตริง s2

10) เมื่อกู้หาเปรียบเทียบการเท่ากันของสตริงโดยไม่สนใจตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็ก

`<สตริง>.equalsIgnoreCase(<สตริง>)`

คืนค่าความจริงที่เป็น true ถ้าสตริงเท่ากัน หรือ false ถ้าสตริงไม่เท่ากัน โดยไม่สนใจตัวอักษรพิมพ์ใหญ่พิมพ์เล็ก

ตัวอย่างเช่น

```
String s1 = "E-mail";
String s2 = "e-mail";
boolean flag = s1.equalsIgnoreCase(s2); //flag = true
```

11) เมท็อดเปรียบเทียบลำดับของสตริง (ใช้ค่า Unicode ในการเปรียบเทียบ)

<สตริง>.compareTo(<สตริง>)

คืนค่า 0 เมื่อสตริงทั้งสองเท่ากัน

เปรียบเทียบรหัส

Unicode ของอัก-

ขระทีละตัวเริ่ม

จากซ้ายไปขวา

ตัวอย่างเช่น

String s1 = "E-mail";

String s2 = "e-mail";

int flag = s1.compareTo(s2); //flag = -32

คืนค่าที่น้อยกว่า 0 เมื่อสตริงตัวแรกมาก่อนสตริงตัวหลัง

คืนค่าที่มากกว่า 0 เมื่อสตริงตัวแรกมาหลังสตริงตัวหลัง

โจทย์ข้อที่ 1 [ระดับง่าย] จงเติมเต็มคำสั่งการประกาศและกำหนดค่าของตัวแปรต่อไปนี้ให้สมบูรณ์ พร้อมทั้งระบุค่าที่เก็บอยู่ในตัวแปรแต่ละข้อ (20 คะแนน)

```
String a = "I Love Java";
String b = "Chula ";
```

ข้อ	ประเภทตัวแปร	การประกาศและกำหนดค่าตัวแปร	ค่าที่เก็บในตัวแปร
1.		var = b.length();	
2.		var = b.trim().length();	
3.		var = "\t".length();	
4.		var = "\n\t\u0000".length();	
5.		var = b.toUpperCase() + b.toLowerCase();	
6.		var = a.charAt(4);	
7.		var = a.indexOf("v");	
8.		var = a.lastIndexOf("v");	
9.		var = a.indexOf("ava");	
10.		var = a.lastIndexOf("ava");	
11.		var = a.indexOf("java");	
12.		var = a.toLowerCase().indexOf("java");	
13.		var = a.substring(6, a.length()).trim() + b;	
14.		var = a.substring(0, 4);	
15.		var = a.toLowerCase().substring(2);	
16.		var = a.substring(7).equals("java");	
17.		var = "Java".equalsIgnoreCase("java");	
18.		var = ".".equals(".".toUpperCase());	
19.		var = "Jaba".compareTo("Java");	
20.		var = "a".compareTo(a.charAt(8));	

3. เมท็อดอื่นๆ ที่เกี่ยวกับสตริง

1) การแปลงสตริงให้เป็นตัวเลขจำนวนเต็ม (String to Integer)

Integer.parseInt (<สตริง>)

ตัวอย่างเช่น

```
String price = "500";
int p = Integer.parseInt(price); //p = 500
int len = p.length(); //Error
```

คืนค่าจำนวนเต็มที่มาจากสตริงที่เป็นตัวเลขจำนวนเต็ม (ห้ามมีอักขระอื่นๆ ที่ไม่ใช่ตัวเลขปะปน)

2) การแปลงสตริงให้เป็นตัวเลขจำนวนจริง (String to Double)

Double.parseDouble (<สตริง>)

ตัวอย่างเช่น

```
String price = "500.0";
double p = Double.parseDouble(price); //p = 500.0
```

คืนค่าจำนวนจริงที่มาจากสตริงที่เป็นตัวเลขจำนวนจริงหรือจำนวนเต็ม

3) การแปลงตัวเลขจำนวนเต็มให้เป็นสตริง (Integer to String)

Integer.toString (<จำนวนเต็ม>)

ตัวอย่างเช่น

```
int price = 500;
String p = Integer.toString(price); //p = "500"
```

คืนค่าสตริงที่มีค่าเหมือนกับตัวเลขจำนวนเต็มที่เรา (หรือใช้วิธี <จำนวนเต็ม> + "")

4) การแปลงตัวเลขจำนวนจริงให้เป็นสตริง (Double to String)

Double.toString (<จำนวนจริง>)

ตัวอย่างเช่น

```
double price = 500.0;
String p = Double.toString(price); //p = "500.0"
```

คืนค่าสตริงที่มีค่าเหมือนกับตัวเลขจำนวนจริงที่เรา (หรือใช้วิธี <จำนวนจริง> + "")

โจทย์ข้อที่ 2 [ระดับง่าย] จงเติมเต็มคำสั่งการประกาศและกำหนดค่าของตัวแปรต่อไปนี้ให้สมบูรณ์ พร้อมทั้งระบุค่าที่เก็บอยู่ในตัวแปรแต่ละข้อ (10 คะแนน)

```
String a = "2000";
String b = "1000.000";
```

```
int n = 2000;
double m = 1000.000;
```

ข้อ	ประเภทตัวแปร	การประกาศและกำหนดค่าตัวแปร	ค่าที่เก็บในตัวแปร
1.		var = Integer.parseInt(a);	
2.		var = Integer.parseInt(b);	
3.		var = Double.parseDouble(a);	
4.		var = Double.parseDouble(b);	
5.		var = Integer.toString(n);	
6.		var = Integer.toString(m);	
7.		var = Double.toString(n);	
8.		var = Double.toString(m);	

ข้อ	ประเภทตัวแปร	การประกาศและกำหนดค่าตัวแปร	ค่าที่เก็บในตัวแปร
*9.		<code>var = Integer.toString(n) == (n + "");</code>	
10.		<code>var = Integer.toString(n).equals(n + "");</code>	

โจทย์ข้อที่ 3 [ระดับง่าย] จงเขียนส่วนหนึ่งของโปรแกรมจากข้อกำหนดต่อไปนี้ (8 คะแนน)

```
String p = "A1234567Java";
String r = "F6543210Jaba";
```

1) เปลี่ยนค่าในตัวแปร `p` และ `r` ให้เป็นตัวพิมพ์ใหญ่ทั้งหมดโดยห้ามประกาศตัวแปรใหม่ (2 คะแนน)

2) ประกาศตัวแปรตรรกะ `y` เพื่อเก็บผลการเปรียบเทียบอักขระตัวที่ 5 ของตัวแปร `p` และ `r` โดยสมมุติให้ตัวแปรทั้งสองมีความยาวเกิน 5 อักขระ และห้ามประกาศตัวแปรอื่นๆ เพิ่มเติม (2 คะแนน)

3) ประกาศตัวแปรจำนวนจริง `m` เพื่อเก็บผลการเปลี่ยนสตริงตั้งแต่ตัวที่ 2 จนถึงตัวที่ 6 ของตัวแปร `p` ให้เป็นตัวเลขจำนวนจริง โดยห้ามประกาศตัวแปรอื่นๆ เพิ่มเติม (2 คะแนน)

โจทย์ข้อที่ 4 [ระดับง่าย] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อรับคำ (Words) จากแป้นพิมพ์เข้ามาทีละคำ โดยให้โปรแกรมนับคำไปเรื่อยๆ จนกว่าผู้ใช้จะพิมพ์คำว่า "Stop" (ไม่สนใจอักขระพิมพ์ใหญ่พิมพ์เล็ก) จึงจะจบการวนซ้ำ พร้อมทั้งแสดงผลลัพธ์ของคำทุกคำค้นด้วยช่องว่างออกจากจอภาพ ดังตัวอย่างต่อไปนี้ (10 คะแนน)

```
Enter Word: I
Enter Word: love
Enter Word: JAVA
Enter Word: very
Enter Word: much
Enter Word: sToP
I love JAVA very much
```

โจทย์ข้อที่ 5 [ระดับปานกลาง] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อรับประโยคจากแป้นพิมพ์หนึ่งประโยค แล้วทำการตัดช่องว่างทุกช่องในประโยคนั้นออกไป โดยให้เหลือเพียงอักขระอื่นๆ ที่ไม่ใช่ช่องว่าง พร้อมทั้งแสดงผลลัพธ์ที่ได้ออกทางจอภาพดังตัวอย่างต่อไปนี้ (10 คะแนน)

Enter Sentence: We love JAVA
WeloveJAVA

โจทย์ข้อที่ 6 [ระดับปานกลาง] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อรับประโยคจากแป้นพิมพ์หนึ่งประโยคแล้วทำการกลับ (Reverse) ประโยคนั้นเสียใหม่ โดยเรียงอักขระทุกตัวจากหลังมาหน้า (ขวาไปซ้าย) พร้อมทั้งแสดงผลลัพธ์ที่ได้ออกทางจอภาพดังตัวอย่างต่อไปนี้ (10 คะแนน)

Enter Sentence: We love JAVA
AVAJ evol eW

โจทย์ข้อที่ 7 [ระดับปานกลาง] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อรับชื่อ-นามสกุลที่แบ่งด้วยช่องว่างอย่างน้อยหนึ่งช่องขึ้นไปเก็บไว้ในตัวแปร fullName หลังจากนั้นให้ทำการแยกชื่อและนามสกุลออกเป็น 2 ส่วนคือ ส่วนแรก “ชื่อ” ให้เก็บไว้ในตัวแปร firstName พร้อมแสดงผลเป็นตัวอักษรตัวพิมพ์ใหญ่ และส่วนที่สอง “นามสกุล” ให้เก็บไว้ในตัวแปร lastName พร้อมแสดงผลเป็นตัวอักษรตัวพิมพ์เล็ก แต่ถ้การรับชื่อ-นามสกุลจากแป้นพิมพ์ไม่ถูกต้องจะแสดงข้อความว่า “Incorrect Name” (10 คะแนน)

```

Full Name: Wayne Rooney
First Name : WAYNE
Last Name : rooney

Full Name: Wayne Rooney
First Name : WAYNE
Last Name : rooney

Full Name: Rooney
Incorrect Name
  
```

โจทย์ข้อที่ 8 [ระดับปานกลาง] จงเขียนส่วนของโปรแกรมภาษาจาวาเพื่อตัดค่า Title และ URL ในโค้ดเว็บเพจที่เขียนด้วยภาษาเอชทีเอ็มแอล (HTML) ต่อไปนี้ (8 คะแนน)

```

1 <html>
2   <head>
3     <title>First Web Page</title>
4   </head>
5   <body>
6     <a href="http://www.javachula.co.cc">JavaChula</a>
7   </body>
8 </html>
  
```

Annotations: Title points to the title tag, URL points to the href attribute.

- 1) ประกาศตัวแปร s1 เพื่อเก็บโค้ดเว็บเพจในบรรทัดที่ 3 และตัวแปร s2 เพื่อเก็บโค้ดเว็บเพจในบรรทัดที่ 6 (2 คะแนน)

- 2) ประกาศตัวแปร title เพื่อเก็บค่า Title ที่ได้จากการตัดโค้ดเว็บเพจในตัวแปร s1 (3 คะแนน)

3) ประกาศตัวแปร `ur1` เพื่อเก็บค่า URL ที่ได้จากการตัดโค้ดเว็บเพจในตัวแปร `s2` (3 คะแนน)

โจทย์ข้อที่ 9 [ระดับปานกลาง] จงเขียนส่วนของโปรแกรมภาษาจาวาเพื่อตัดค่าของวัน เดือน ปี จากตัวแปรสตริง `a` ที่รับค่าวัน เดือน ปี จากแป้นพิมพ์ไว้เรียบร้อยแล้วในรูปแบบของ "1/12/2012" หรือ "13/6/2010" เป็นต้น พร้อมทั้งแสดงผลลัพธ์ของวัน เดือน ปี ออกทางจอภาพที่ละบรรทัด (6 คะแนน)

โจทย์ข้อที่ 10 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อรับข้อความหนึ่งข้อความจากแป้นพิมพ์ เพื่อนำมาตรวจสอบว่าข้อความดังกล่าวเป็นข้อความแบบพาลินโดรม (Palindrome) หรือไม่ โดยพาลินโดรมคือข้อความที่อ่านจากหน้าไปหลังเหมือนกับอ่านจากหลังไปหน้า เช่น "level", "2552", "race car", "I prefer pi" เป็นต้น ทั้งนี้จะไม่สนใจช่องว่างและตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กในการอ่านแต่อย่างใด (10 คะแนน)

Text: I prefer pi
It is palindrome

Text: I prefer Java
It is not palindrome

โจทย์ข้อที่ 11 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อรับข้อความที่ประกอบไปด้วยตัวเลขและตัวอักษรปะปนกัน โดยไม่จำกัดความยาว จากนั้นให้ทำการตรวจสอบอักขระทีละตัวว่าเป็นตัวเลขหรือตัวอักษร ถ้าเป็นตัวเลขจะนำมาหา [1] ค่ามากที่สุด [2] ค่าน้อยที่สุด และ [3] ค่าเฉลี่ย ถ้าเป็นตัวอักษรจะนับจำนวนอักขรทั้งหมดที่พบ พร้อมทั้งแสดงผลลัพธ์ทั้ง 4 ค่าออกทางจอภาพ (ห้ามใช้เมธอดจากคลาส Math) (10 คะแนน)

```
import java.util.Scanner;
public class NumberAndCharacter {
    public static void main(String[] args) {
```

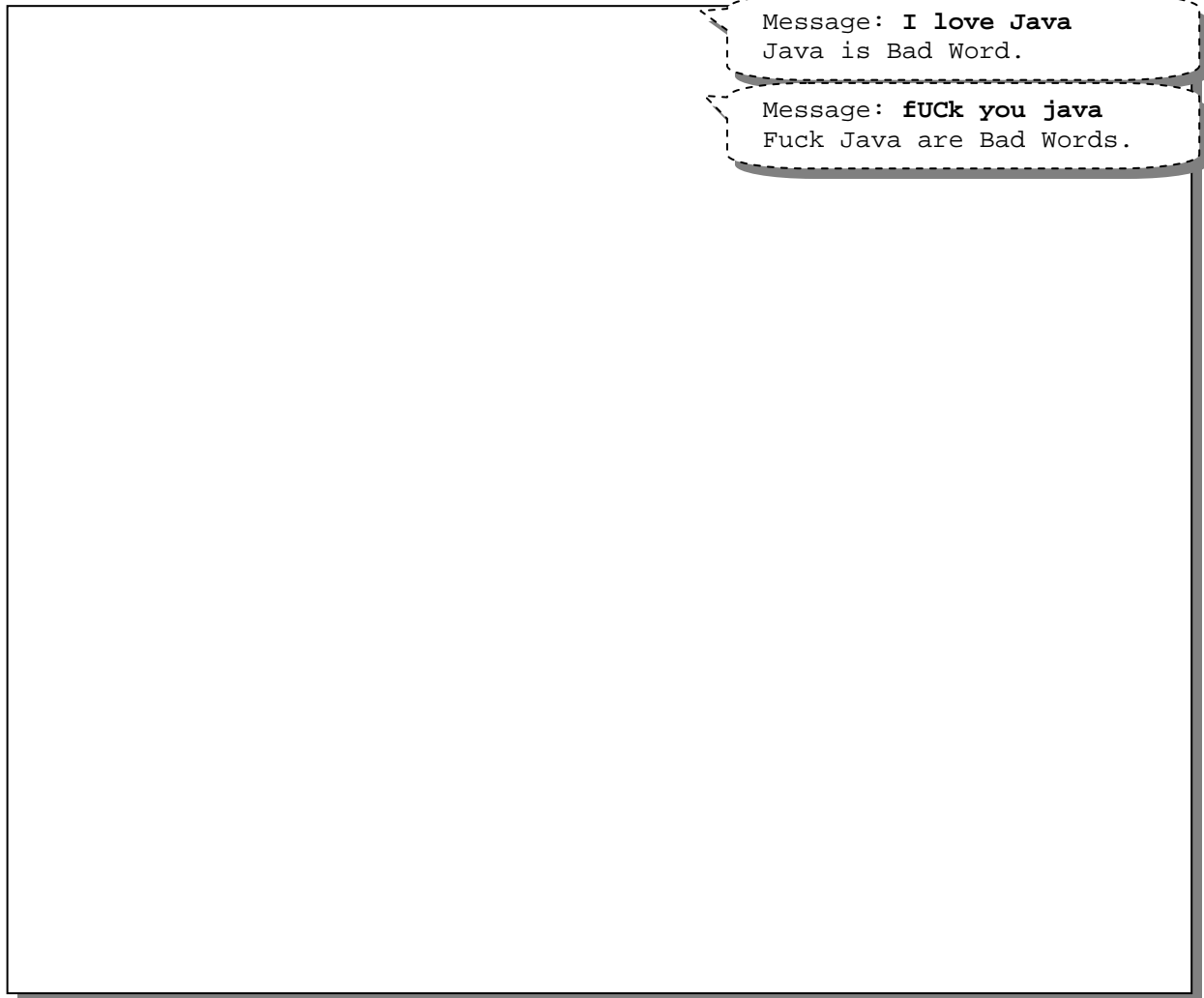
```
STRING: loveU2java37A1cu5
MAX VALUE: 7
MIN VALUE: 1
AVERAGE VALUE (18/5): 3.6
TOTAL CHARACTER: 12
```

```
    } //End of main
} //End of class
```

โจทย์ข้อที่ 12 [ระดับยาก] ในการสร้างโปรแกรมสนทนาผ่านทางอินเทอร์เน็ตของบริษัทลานเกียร์จำกัด ชื่อว่าโปรแกรม “เกรียนแชท” (Noob Chat) ซึ่งต้องตรวจสอบข้อความที่ผู้ใช้พิมพ์เข้ามาสนทนากัน ว่ามีคำหยาบ ซึ่งได้แก่ “Shit” “Fuck” และ “Java” อยู่หรือไม่ ซึ่งไม่ว่าเป็นตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กก็สามารถตรวจสอบได้ทุกกรณี ถ้าตรวจสอบแล้วไม่มีคำหยาบปรากฏ ให้แสดงข้อความนั้นบนจอภาพ ถ้าไม่เช่นนั้นให้แสดงข้อความว่ามีคำหยาบบนจอภาพ ดังตัวอย่างต่อไปนี้ (15 คะแนน)

```
import java.util.Scanner;
public class NoobChat {
    public static void main(String[] args) {
```

```
Message: Hello F
Hello F
```



```
} //End of main
} //End of class
```

โจทย์ข้อที่ 13 [ระดับเทพ] จากที่โปรแกรม “เกรียนแชท” (Noob Chat) ของบริษัทลานเกียร์จำกัดซึ่งเขียนโดยนิสิตวิศวะฯ ปี 1 ได้ถูกนำไปใช้งานในระบบจริงแล้วปรากฏว่า ยังมีบั๊ก (Bug) เกิดขึ้นอีกมาก นั่นคือไม่สามารถรับข้อความที่มากกว่า 1 บรรทัดได้ และไม่สามารถนับจำนวนคำหยาบที่ปรากฏในข้อความได้ ดังนั้นบริษัทลานเกียร์จึงส่งโปรแกรมดังกล่าวกลับมายังนิสิตผู้พัฒนา เพื่อปรับปรุงให้สมบูรณ์และเปลี่ยนชื่อใหม่เป็นโปรแกรม “เกรียนเทพแชท” (God Noob Chat) โดยมีรายละเอียดดังนี้ (20 คะแนน)

- 1) รับข้อความจากแป้นพิมพ์ได้ไม่จำกัดบรรทัด และจบเมื่อพิมพ์จุด (Dot) ติดต่อกัน 3 ครั้ง
- 2) ตรวจสอบข้อความที่พิมพ์เข้ามาว่ามีคำหยาบอยู่หรือไม่ ซึ่งได้แก่ “Shit” “Fuck” และ “Java” ไม่ว่าจะเป็นตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กก็สามารถตรวจสอบได้ทุกกรณี และไม่ว่าคำหยาบจะถูกแบ่งด้วยช่องว่างเช่น “F U C K” ก็สามารถตรวจสอบได้เช่นกัน
- 3) นับจำนวนคำหยาบแต่ละคำที่ปรากฏในข้อความและแสดงจำนวนคำหยาบที่ตรวจพบบนจอภาพ
- 4) ถ้าตรวจสอบแล้วไม่มีคำหยาบปรากฏให้แสดงข้อความนั้นบนจอภาพ ถ้าไม่เช่นนั้นให้แสดงข้อความว่ามีคำหยาบบนจอภาพ

```
import java.util.Scanner;
public class GodNoobChat {
    public static void main(String[] args) {
```

Message:

F U C K you JaVA
JA V a ShiTTtt FU
C King

...

Count "Shit": 1

Count "Fuck": 2

Count "Java": 2

Shit Fuck Java are bad words.

```
} //End of main  
} //End of class
```