

Java Structure Programming

การเขียนโปรแกรมแบบโครงสร้างด้วยภาษาจาวา

จาวาฉบับพื้นฐาน ด้วยการเขียนโปรแกรมแบบโครงสร้าง

❖ ความรู้เบื้องต้นของภาษาจาวา

ตัวอย่างอ่านฟรี

- ❖ สตริง
- ❖ เมธอดและการเรียกใช้คลาส
- ❖ การดักจับข้อผิดพลาด
- ❖ เพิ่มข้อมูล
- ❖ ความรู้เบื้องต้นของการเขียนโปรแกรมเชิงวัตถุ

โดย เซาวลิต ชันคำ

สารบัญ

หน้า

คำนำ.....	(1)
สารบัญ.....	(3)
สารบัญภาพ.....	(9)
สารบัญตาราง.....	(13)
บทที่ 1 การเขียน โปรแกรมคอมพิวเตอร์.....	1
โปรแกรมคอมพิวเตอร์.....	1
โปรแกรมภาษาคอมพิวเตอร์.....	3
ยุคของโปรแกรมภาษาคอมพิวเตอร์.....	5
การเขียน โปรแกรมคอมพิวเตอร์.....	10
รูปแบบพื้นฐานของการเขียน โปรแกรม.....	13
สรุป.....	22
แบบฝึกหัดท้ายบท.....	23
เอกสารอ้างอิง.....	25
บทที่ 2 โปรแกรมภาษาจาวา.....	27
ความรู้เบื้องต้นของภาษาจาวา.....	27
การติดตั้งภาษาจาวา.....	33
โครงสร้างและรูปแบบพื้นฐานของภาษาจาวา.....	37
สรุป.....	45
แบบฝึกหัดท้ายบท.....	46
เอกสารอ้างอิง.....	47
บทที่ 3 คำสั่งแสดงผล.....	51
การแสดงผลมาตรฐาน.....	51
การจัดรูปแบบการแสดงผล.....	59
การจัดรูปแบบเลขทศนิยม.....	66
สรุป.....	68
แบบฝึกหัดท้ายบท.....	69

เอกสารอ้างอิง.....	71
บทที่ 4 คำสั่งรับข้อมูล.....	73
หลักการนำเข้าข้อมูลเบื้องต้น.....	73
ตัวแปรและการประกาศตัวแปร	75
การนำเข้าข้อมูลด้วย BufferedReader	79
การนำเข้าข้อมูลด้วย Scanner	88
การนำเข้าข้อมูลด้วย JOptionPane.....	91
สรุป.....	93
แบบฝึกหัดท้ายบท	95
เอกสารอ้างอิง.....	97
บทที่ 5 ตัวดำเนินการ.....	99
ความหมายและประเภทของตัวดำเนินการ	99
ตัวดำเนินการกำหนดค่า	100
ลำดับขั้นตอนดำเนินการ	102
ตัวดำเนินการทางคณิตศาสตร์	103
ตัวดำเนินการเปรียบเทียบความสัมพันธ์และตรรกะ.....	123
ตัวดำเนินการบิตไวส์.....	127
สรุป.....	132
แบบฝึกหัดท้ายบท	133
เอกสารอ้างอิง.....	135
บทที่ 6 คำสั่งเงื่อนไข.....	137
ความหมายและความสำคัญของคำสั่งเงื่อนไข.....	137
คำสั่ง if.....	138
คำสั่ง switch	153
สรุป.....	160
แบบฝึกหัดท้ายบท	161
เอกสารอ้างอิง.....	163

บทที่ 7	คำสั่งวนซ้ำ	165
	ความหมายและประเภทของคำสั่งวนซ้ำ.....	165
	คำสั่ง for	166
	คำสั่ง while	174
	คำสั่ง do...while	179
	สรุป	186
	แบบฝึกหัดท้ายบท	187
	เอกสารอ้างอิง	189
บทที่ 8	อาร์เรย์	191
	ความหมายและความสำคัญของอาร์เรย์	191
	อาร์เรย์ 1 มิติ	192
	อาร์เรย์ 2 มิติ	198
	อาร์เรย์ 3 มิติ	205
	สรุป	211
	แบบฝึกหัดท้ายบท	212
	เอกสารอ้างอิง	213
บทที่ 9	สตริง	215
	ความรู้พื้นฐานเกี่ยวกับสตริง	215
	การดำเนินการกับสตริง	219
	สรุป	238
	แบบฝึกหัดท้ายบท	239
	เอกสารอ้างอิง	241
บทที่ 10	เมฆอดและการเรียกใช้คลาส	243
	ความหมายและและความสำคัญของเมฆอด	243
	ตำแหน่งการวางเมฆอดในโปรแกรม	247
	เมฆอดแบบไม่คืนค่า	248
	เมฆอดแบบคืนค่า	262
	การใช้งานคลาส	266

	หน้า
สรุป.....	273
แบบฝึกหัดท้ายบท	274
เอกสารอ้างอิง.....	275
บทที่ 11 การดักจับข้อผิดพลาด	277
ความหมายและความสำคัญของการดักจับข้อผิดพลาด	277
รูปแบบของการดักจับ	280
การสร้างการดักจับข้อผิดพลาด.....	292
การจัดข้อผิดพลาด	294
ข้อผิดพลาดในแพคเกจที่ใช้งานเป็นประจำ	300
สรุป.....	306
แบบฝึกหัดท้ายบท	307
เอกสารอ้างอิง.....	309
บทที่ 12 เพิ่มข้อมูล.....	311
ความสำคัญและประเภทของเพิ่มข้อมูล	311
เท็กซ์ไฟล์	313
การอ่านเพิ่มเท็กซ์.....	319
การเขียนเพิ่มด้วย Printer Writer	321
เพิ่มแบบสุ่ม.....	325
การบริหารจัดการเพิ่ม	337
สรุป.....	340
แบบฝึกหัดท้ายบท	341
เอกสารอ้างอิง.....	343
บทที่ 13 ความรู้เบื้องต้นของการเขียน โปรแกรมเชิงวัตถุ	345
แนวคิดการเขียน โปรแกรมด้วยภาษาเชิงวัตถุ.....	345
ส่วนประกอบเชิงวัตถุ.....	348
คลาส.....	351
การสร้างคลาสและการสร้างวัตถุ.....	354
การสืบทอดคุณสมบัติ	359

การโอเวอร์โหลด.....	367
การโอเวอร์ไรด์.....	369
สรุป	371
แบบฝึกหัดท้ายบท.....	372
เอกสารอ้างอิง.....	375
บรรณานุกรม.....	377

บทที่ 3

คำสั่งแสดงผล

การเขียนโปรแกรมคอมพิวเตอร์ เริ่มจากการเขียนรหัสต้นฉบับให้ถูกต้องตามรูปแบบที่กำหนดของไวยากรณ์ของโปรแกรมภาษา หลังจากนั้นคอมไพล์โปรแกรมเพื่อแปลคำสั่งจากรหัสต้นฉบับให้อยู่ในรูปแบบของคอมพิวเตอร์เข้าใจและหากไม่มีข้อผิดพลาด โปรแกรมจะสามารถรันได้ทันที นอกจากนี้ผู้เขียนโปรแกรมต้องมีความรู้ด้านการวิเคราะห์และออกแบบโปรแกรมตามแนวทางของอัลกอริทึม (Algorithm) เพื่อออกแบบและเขียนโปรแกรมให้ได้ตามที่ผู้ใช้งานต้องการ การฝึกเขียนโปรแกรมที่ง่ายที่สุด คือ การเริ่มสั่งให้คอมพิวเตอร์แสดงผลทางจอภาพให้ได้ผลดังที่ผู้เขียนต้องการ ด้วยคำสั่งแสดงผลพื้นฐาน บทนี้แนะนำคำสั่งที่ใช้สำหรับการแสดงผลเบื้องต้น เพื่อให้ผู้เรียนได้มีความรู้เรื่องรูปแบบการเขียนโปรแกรม การแสดงผลออกทางจอภาพตามที่ต้องการ โดยแบ่งออกเป็นคำสั่งแสดงผลด้วยคำสั่งพื้นฐาน และการจัดรูปแบบข้อมูลให้ได้ดังที่ต้องการ เช่น จัดรูปแบบตัวเลข จัดรูปแบบข้อความ ข้อมูลวันที่ เป็นต้น

การแสดงผลมาตรฐาน

คำสั่งสำหรับแสดงผลของภาษาจาวาโดยทั่วไปจะใช้ `System.out`. และตามด้วยชื่อของเมธอดแสดงผล ซึ่งข้อมูลที่แสดงผลนั้นจะเป็นข้อมูลที่เป็นทั้งข้อความ ตัวเลขในรูปแบบต่าง ๆ และการแปลงวันที่เป็นข้อความเพื่อแสดงผล โดยที่คำสั่ง `.out` นี้จะอยู่ในคลาส `System` (ธีรวัฒน์ ประกอบผล, 2552, หน้า 23) และธีรวัฒน์ ประกอบผล (2553, หน้า 29) ก็ยังกล่าวอีกว่า การแสดงผลข้อมูลทางจอภาพมักเป็นเมธอดแรกที่นักเขียนโปรแกรมจะต้องศึกษาในภาษาจาวาจะใช้เมธอด `print` และ `println` ในการแสดงผลโดยเมธอดทั้งสองนี้จะถูกบรรจุอยู่ในออบเจกต์ที่ชื่อ `out` ซึ่งออบเจกต์นี้จะถูกบรรจุอยู่ในคลาส `System` การเรียกใช้เมธอดที่บรรจุอยู่จะใช้เครื่องหมายจุด (`.`) ในการอ้างอิง

ส่วนพนิดา พานิชกุล (2554, หน้า 49) กล่าวไว้ในหมายเหตุว่า “`System.out.println(“Hi this is my first program”)`” เป็นคำสั่งที่ใช้พิมพ์ข้อความออกจอภาพ โดยคำว่า “`System`” จัดเป็นคลาส ส่วนคำว่า “`out`” จัดเป็นออบเจกต์ และ “`println`” จัดเป็นเมธอด นับว่าเป็นรูปแบบหนึ่งของการเขียนประโยค

คำสั่งของจาวา สรุปได้ว่ารูปแบบการสั่งจะกำหนดเป็น class.object.method

ริชาร์ด เอฟ. แรพโพสา (Richard F. Raposa, 2003, p.13) อธิบายคำสั่ง System.out.println(“Hello, World”) ไว้ว่า ข้อความ “Hello, World” จะปรากฏในคอมมานด์พรอมต์ หรือในกรณีนี้จะปรากฏในการแสดงผลมาตรฐาน คำสั่ง System.out ใช้แทนการแสดงผลมาตรฐานที่อยู่ ณ ที่โปรแกรมนั้นรันอยู่

ฮาร์เวย์ เอ็ม. เดทเทล (Harvey M. Deitel, 2004, p.60) อธิบายว่า System.out รู้จักในนามของ ออบเจกต์แสดงผลมาตรฐานโดย System.out อนุญาตให้โปรแกรมของจาวาแสดงสตริงหรือข้อความ อื่น ๆ ในหน้าต่างคำสั่งจากตำแหน่งที่มีการรัน โปรแกรมนั้น ๆ

ดังนั้นจึงกล่าวได้ว่าคำสั่งแสดงผลมาตรฐานที่ใช้สำหรับภาษาจาวานั้นจะใช้งานคลาส System ด้วยการเรียกใช้ออบเจกต์ out และเรียกใช้งานผ่านเมธอด print และ println โดยใช้จุด (.) เป็นตัวแบ่ง การเรียกใช้งาน เช่น System.out.println เป็นต้น ต่อไปนี้เป็นการแสดงการใช้งานเมธอดทั้งสองโดยละเอียดและขณะเดียวกันจะนำเสนอตัวอย่างการใช้งานของเมธอดดังกล่าวในรูปแบบแตกต่างกัน

คำสั่ง System.out.print

พอล เดทเทล และ ฮาร์ลีย์ เดทเทล (Paul Deitel and Harvey Deitel, 2010, p.62) กล่าวว่า System.out.print แสดงข้อมูลออก ณ ที่อยู่ของค่าคงที่หรือตำแหน่งที่เคอร์เซอร์อยู่ในปัจจุบันหลังจากอักขระสุดท้ายถูกแสดง

คำสั่ง System.out.print นี้สามารถแสดงผลได้ทั้งการผ่านค่าพารามิเตอร์ที่เป็นออบเจกต์ ข้อความ ข้อมูลทางตรรกะ อักขระ และตัวเลขประเภทต่าง ๆ รายละเอียดแสดงดังภาพที่ 3.1

● print (Object obj)	void
● print (String s)	void
● print (boolean b)	void
● print (char c)	void
● print (char[] s)	void
● print (double d)	void
● print (float f)	void
● print (int i)	void
● print (long l)	void

ภาพที่ 3.1 แสดงคำสั่ง System.out.print

จากภาพเมธอด print มีรูปแบบการเรียกใช้งานที่แตกต่างกัน โดยที่

- print(Object obj) เป็นการเรียกใช้งานโดยนำค่าออบเจกต์ผ่านเข้าไปเพื่อเรียกใช้งาน เช่น หากมีออบเจกต์ที่ชื่อว่า showName อาจเรียกใช้ดังนี้ print(showName); เป็นต้น

- print(String s); เป็นการเรียกใช้งานโดยการผ่านค่าสตริงเข้าไปเพื่อแสดงผล เช่น print(“ทดสอบ”); การแสดงผลจะนำเอาข้อความ “ทดสอบ” ออกมาแสดงผลเป็นต้น

- print(boolean b) เป็นการเรียกใช้งานโดยการผ่านค่าบูลีนเข้าไปซึ่งมีค่าเป็นแบบ true/false ตัวอย่างเช่น print(true); โปรแกรมพิมพ์ค่า true ออกมาเป็นต้น

- print(char c)/print(char[] s) เป็นการเรียกใช้งานโดยผ่านค่าอักขระเพียงตัวเดียวหรือสายอักขระเข้าไปแสดงผลก็ได้ ตัวอย่างเช่น หากมีตัวแปร char c= ‘ก’; สามารถสั่งแสดงได้ดังนี้ print(c); ซึ่งโปรแกรมจะแสดงอักขระ ‘ก’ ทางจอภาพเป็นต้น

- print(double d)/print(float f)/print(int i)/print(long l) เป็นการเรียกใช้งานโดยการผ่านค่าตัวเลขในรูปแบบที่แตกต่างกัน ทั้งแบบจำนวนเต็ม ทศนิยม แบบแตกต่างกัน โดยรองรับข้อมูลตามโครงสร้างข้อมูลของภาษาจาวา เช่น หากมีข้อมูล double a=1234.45; สามารถสั่งพิมพ์ออกมาได้ดังนี้ print(a) ซึ่งผลการแสดงจะได้ข้อมูลที่เป็นสตริงในรูปแบบ 1234.45 ทางจอภาพเป็นต้น

ต่อไปนี้จะแสดงตัวอย่างของโปรแกรมด้วยการเรียกใช้งานคำสั่ง System.out.print ในรูปแบบแตกต่างกันดังที่อธิบายไปแล้ว

ตัวอย่างที่ 3.1 จงเขียนโปรแกรมแสดงข้อความ My name is Praram บนจอภาพ 1 ครั้ง

รหัสต้นฉบับ:

(1)	public class Ex_3_1 {	//เริ่มต้นคลาส
(2)	public static void main(String argv[])	//เมธอดหลัก
(3)	{	//เริ่มคำสั่งของเมธอดหลัก
(4)	System.out.print("My name is Praram");	//คำสั่งแสดงผล
(5)	}	//จบส่วนคำสั่งของเมธอดหลัก
(6)	}	//จบคลาส

ผลการรันโปรแกรม จะแสดงข้อความ My name is Praram บนจอภาพ

ตัวอย่างที่ 3.2 จงเขียน โปรแกรมแสดงตัวเลข int x=20, double y=50.0

รหัสต้นฉบับ:

```
(1) public class Ex_3_2 { //เริ่มต้นคลาส
(2)     public static void main(String argv[]) //เมธอดหลัก
(3)     { //เริ่มคำสั่งของเมธอดหลัก
(4)         int x=20; //ประกาศตัวแปรจำนวนเต็มเพื่อแสดงแสดงผล
(5)         double y=50.0; //ประกาศตัวแปรทศนิยมเพื่อแสดงแสดงผล
(6)         System.out.print(x); //แสดงผลตัวแปร x
(7)         System.out.print(y); //แสดงผลตัวแปร y
(8)     } //จบส่วนคำสั่งของเมธอดหลัก
(9) } //จบคลาส
```

ผลการรันโปรแกรม แสดงข้อมูล ดังกล่าว ติดกัน 2050.0 ต่อกัน

ตัวอย่างที่ 3.3 จงเขียน โปรแกรมแสดงข้อความ ชื่อ ,สกุล ,ที่อยู่ของตนเองบนจอภาพ

รหัสต้นฉบับ :

```
(1) public class Ex_3_3 { //เริ่มต้นคลาส
(2)     public static void main(String argv[]) //เมธอดหลัก
(3)     { //เริ่มคำสั่งของเมธอดหลัก
(4)         //ส่วนการแสดงผลตามที่โจทย์กำหนด
(5)         System.out.print("ชื่อ:ชาวลิต");
(6)         System.out.print("สกุล:จันทร์คำ");
(7)         System.out.print("ที่อยู่:มหาวิทยาลัยราชภัฏราชนครินทร์ จ.ฉะเชิงเทรา");
(8)     } //จบส่วนคำสั่งของเมธอดหลัก
(9) } //จบคลาส
```

ผลการรันของโปรแกรมนี้อาจจะแสดงชื่อ นามสกุลและที่อยู่ ดังนี้

ชื่อ:ชาวลิตสกุล:จันทร์คำที่อยู่:มหาวิทยาลัยราชภัฏราชนครินทร์ จ.ฉะเชิงเทรา

โปรแกรมนี้แสดงข้อมูลต่อกัน หากต้องการให้แสดงขึ้นบรรทัดใหม่ให้ใช้ \n ซึ่งเป็นรหัสหลุด

พีน (Escape sequence) ซึ่งจะแสดงในตัวอย่างที่ 3.4 ในหัวข้อต่อไป หรือเรียกใช้งานโดยใช้คำสั่ง `System.out.println` เป็นต้น

คำสั่ง `System.out.println`

ฮาร์เวย์ เอ็ม. เดทเทิล (Harvey M. Deitel, 2004, p.60) กล่าวว่า เมธอด `System.out.println` แสดงหรือพิมพ์ที่กซ์เป็นบรรทัดในหน้าต่างคำสั่งเมื่อ `System.out.println` สิ้นสุดคำสั่ง มันจะกำหนดตำแหน่งแสดงผลโดยอัตโนมัติ (ตำแหน่งที่อยู่ถัดจากการแสดงผลที่ถูกแสดงผลอยู่) และจะแสดงเป็นบรรทัดต่อไปในหน้าต่างแสดงผล

สจิวัด เรกเกส และ มาร์ที สเต็ปป์ (Stuart Reges and Marty Stepp, 2008, p.15) กล่าวว่า หนึ่งในประโยคคำสั่งที่ง่ายและเป็นพื้นฐานที่สุดก็คือ `System.out.println` ซึ่งถูกใช้จัดการบรรทัดของข้อความสำหรับแสดงผลออกทางอุปกรณ์แสดงผลมาตรฐาน

คำสั่ง `System.out.println` นี้สามารถเรียกใช้งานได้คล้ายกันกับคำสั่ง `System.out.print` แต่ต่างกันที่เมื่อแสดงผลเสร็จแล้วจะขึ้นบรรทัดใหม่และเลื่อนเคอร์เซอร์ไปยังต้นบรรทัดให้โดยอัตโนมัติ ลักษณะการใช้งานของคำสั่งแสดงดังภาพที่ 3.2

● <code>println()</code>	<code>void</code>
● <code>println(Object x)</code>	<code>void</code>
● <code>println(String x)</code>	<code>void</code>
● <code>println(boolean x)</code>	<code>void</code>
● <code>println(char x)</code>	<code>void</code>
● <code>println(char[] x)</code>	<code>void</code>
● <code>println(double x)</code>	<code>void</code>
● <code>println(float x)</code>	<code>void</code>
● <code>println(int x)</code>	<code>void</code>
● <code>println(long x)</code>	<code>void</code>

ภาพที่ 3.2 แสดงคำสั่ง `System.out.println`

จากภาพ 3.2 อธิบายได้ว่า การเรียกใช้งาน `System.out.println` นั้นสามารถเรียกใช้งานได้แบบ `System.out.println()`; โดยไม่ต้องผ่านค่าใด ๆ ก็ได้ซึ่งผลลัพธ์จะแสดงบรรทัดว่างนั่นเอง แต่การเรียกใช้งานในลักษณะอื่นเช่น `Object`, `String`, `boolean`, `char`, `char []`, `double`, `float`, `int`, `long` จะมีรูปแบบการแสดงผลเช่นเดียวกันกับคำสั่ง `System.out.print` ซึ่งจะแตกต่างกันก็คือเมื่อแสดงผลเสร็จแล้วเคอร์เซอร์จะถูกกำหนดให้ไปอยู่ในบรรทัดถัดไป ตัวอย่างการใช้งานเช่น ในตัวอย่างที่ 3.3 หากต้องการให้ขึ้น

บรรทัดใหม่ต้องแสดงดังนี้

```
System.out.println("ชื่อ:เซาวลิต");
System.out.println("สกุล:ชั้นคำ");
System.out.println("ที่อยู่:มหาวิทยาลัยราชภัฏราชชนครินทร์ จ.ฉะเชิงเทรา");
```

ตัวอย่างที่ 3.4 จงเขียนโปรแกรมแสดงข้อความ ชื่อ ,สกุล ,ที่อยู่ของตนเองบนจอภาพโดยเว้นบรรทัดในการแสดงด้วย

รหัสต้นฉบับ :

```
(1) public class Ex_3_4 { //เริ่มต้นคลาส
(2)     public static void main(String argv[]) //เมธอดหลัก
(3)     { //เริ่มคำสั่งของเมธอดหลัก
           //ส่วนการแสดงผลตามที่โจทย์กำหนด
(4)         System.out.println("ชื่อ:เซาวลิต");
(5)         System.out.println();
(6)         System.out.println("สกุล:ชั้นคำ");
(7)         System.out.println();
(8)         System.out.println("ที่อยู่:มหาวิทยาลัยราชภัฏราชชนครินทร์ จ.ฉะเชิงเทรา");
(9)     } //จบส่วนคำสั่งของเมธอดหลัก
(10) } //จบคลาส
```

ตัวอย่างที่ 3.5 จงเขียนโปรแกรมแสดงข้อความ ค่าตรรกะ ตัวเลขจำนวนเต็ม ทศนิยม โดยกำหนดให้
String s= "Test String" boolean=true int x=40 float y=500

รหัสต้นฉบับ :

```
(1) public class Ex_3_5 { //เริ่มต้นคลาส
(2)     public static void main(String argv[]) //เมธอดหลัก
(3)     { //เริ่มคำสั่งของเมธอดหลัก
```

```

//ส่วนการประกาศตัวแปรและการกำหนดค่าให้กับตัวแปรเพื่อแสดงผล
(4)   String s="Test String";
(5)   boolean b=true;
(6)   int x=40;
(7)   double y=50.0;

//ส่วนการนำค่าตัวแปรมาแสดงผลตามที่โจทย์กำหนด
(8)   System.out.println("String:"+s);           //แสดงสตริง
(9)   System.out.println("Boolean:"+b );         //แสดงค่าบูลีน
(10)  System.out.println("Integer:"+x);          //แสดงจำนวนเต็ม
(11)  System.out.println("Double:"+y);          //แสดงตัวเลขทศนิยม
(12)  }                                           //จบส่วนคำสั่งของเมธอดหลัก
(13)  }                                           //จบคลาส

```

ลำดับหลุดพ้น(Escape Sequence)

พอล เดทเทล และ ฮาร์เวย์ เดทเทล (Paul Deitel and Harvey Deitel, 2010, p.46) กล่าวว่า \ (backslash) เรียกว่ารหัสหลุดพ้น ใช้แสดงใน System.out.print และ println ซึ่งใช้แสดงอักขระพิเศษ โดยเมื่อปรากฏเครื่องหมายนี้หน้าอักขระ จาาจะรวมอักขระตัวถัดไปมาสร้างรูปแบบรหัสหลุดพ้น เช่น \n หมายถึงการขึ้นบรรทัดใหม่เป็นต้น

ลำดับหลุดพ้น คือ สัญลักษณ์ที่แทรกเข้าไปเพื่อร่วมกับการแสดงผลเพื่อจัดรูปแบบการแสดงผล เช่น สั่งให้ขึ้นบรรทัดใหม่ สั่งให้เลื่อนหน้ากระดาษ เป็นต้น การใช้รหัสเหล่านี้จะเขียนแทรกเข้าไปในคำสั่งแสดงผล เช่น System.out.println("Test\n\t"); โดยที่ \n คือ ลำดับหลุดพ้นเพื่อสั่งให้การแสดงผลขึ้นบรรทัดใหม่ และ \t คือ ลำดับหลุดพ้นเพื่อสั่งให้เลื่อนแท็บในการแสดงผล เป็นต้น รายละเอียดแสดงดังตารางที่ 3.1

ตารางที่ 3.1 แสดงรหัสลำดับหลุดพ้น

Escape Sequence	ความหมาย
\n	การขึ้นบรรทัดใหม่
\t	การสั่งให้เลื่อนแท็บไปหนึ่งแท็บ
\b	การลบอักขระด้านหน้า
\f	การเลื่อนขึ้นหนึ่งหน้า
\r	แทนการกด Enter
\"	การแสดงอักขระ “
'	การแสดงอักขระ ‘
\\	แทนการใช้เครื่องหมาย \
\uDDDD	การแสดงอักขระแบบ Unicode โดยกำหนดให้เป็นรหัสเลขฐานสิบหกจำนวนสี่หลัก

ตัวอย่าง 3.6 จงเขียน โปรแกรมแสดงข้อความ Hello world ที่บริเวณกลางบรรทัด

รหัสต้นฉบับ :

(1)	public class Ex_3_6 {	//เริ่มต้นคลาส
(2)	public static void main(String argv[])	//เมธอดหลัก
(3)	{	//เริ่มคำสั่งของเมธอดหลัก
(4)	System.out.print("\t\t\t\t Hello World");	//แสดงผลด้วยการแทรก \t
(5)	}	//จบส่วนคำสั่งของเมธอดหลัก
(6)	}	//จบคลาส

ผลของการรันโปรแกรมนี้จะแสดงข้อความ Hello world

ตัวอย่างที่ 3.7 จงเขียนโปรแกรมแสดงข้อความ มุมซ้ายของจอภาพแสดงข้อความ My name is Praram มุมขวาแสดง My name is Hero กลางจอภาพแสดงข้อความ Hello world มุมล่างซ้ายแสดงข้อความ My name is Zoro มุมล่างขวาแสดงข้อความ My name is Rambo

รหัสต้นฉบับ :

```
(1) public class Ex_3_7 { //เริ่มต้นคลาส
(2)     public static void main(String argv[]) //เมธอดหลัก
(3)     { //เริ่มคำสั่งของเมธอดหลัก
           //ส่วนการแสดงผลหลักโดยแทรกกรหัสหลุคพื้นเพื่อให้ได้รูปแบบตามโจทย์กำหนด
(4)         System.out.println("My name is Praram \t\t\t\t\t My name is Hero");
(5)         System.out.println("\t\t\t Hello World");
(6)         System.out.println("My name is Zoro \t\t\t\t\t My name is Rambo");
(7)     } //จบส่วนคำสั่งของเมธอดหลัก
(8) } //จบคลาส
```

ผลการรันโปรแกรมจะพิมพ์ข้อความบนจอภาพทั้ง 4 มุมของจอภาพโดยกลางจอภาพจะแสดงข้อความ Hello world ดังนี้

My name is Praram

My name is Hero

Hello World

My name is Zoro

My name is Rambo

การจัดรูปแบบการแสดงผล

การจัดการรูปแบบการแสดงผลสามารถทำได้โดยใช้คำสั่ง System.out.format และ System.out.printf ต่อไปนี้แสดงรายละเอียดการใช้งานคำสั่งทั้งสองนี้ดังนี้

การใช้งานคำสั่ง System.out.format

สำหรับคำสั่งนี้มีรูปแบบดังนี้

System.out.format (“รูปแบบการแสดงผล”, ตัวแปรหรือค่าที่จะแสดงผล);

โดยรูปแบบบังคับเพื่อแสดงผลดังนี้

%d เลขฐานสิบจำนวนเต็ม

%f เลขฐานสิบทศนิยม

%tB วันเวลาโดยกำหนดชื่อเต็มของเดือน ณ พื้นที่ของระบบที่ใช้งาน

%td, %te วันเวลาแบบสั้นด้วยตัวเลขสองตัว โดยที่ %td จะแสดงเลขศูนย์หน้าตัวเลขด้วย หากเป็น %te จะไม่แสดง

%ty, %tY แปลงวันเวลาโดยที่ y หมายถึง ตัวเลข 2 หลักของปี และ Y หมายถึงตัวเลข 4 หลักของปี

%tl แปลงวันเวลาเป็นชั่วโมงใน 12 ชั่วโมงของนาฬิกา

%tM แปลงวันเวลาเป็นนาทีโดยที่ใช้เป็นตัวเลขสองหลักและเพิ่มศูนย์เข้าข้างหน้าด้วยหากจำเป็น

%tp แปลงวันเวลาโดยกำหนดรูปแบบตามพื้นที่ am/pm โดยเป็นตัวยกรตัวเล็ก

%tD แปลงวันเวลาสำหรับวัน โดยใช้รูปแบบ %tm%td%ty

นอกจากการจัดรูปแบบโดยใช้ % แล้ว ในแต่ละประเภทรูปแบบนั้นยังสามารถแทรกส่วนของค่า แฟลก (Flag) ของการแสดงผลได้อีกด้วย ซึ่งมีดังนี้

08 ความกว้างของการแสดงอักขระเป็น 8 ตัวอักษร และเพิ่มเลขศูนย์ด้วยหากไม่ครบ

+ แสดงค่าด้วยเครื่องหมายบวก

- จัดชิดซ้าย

, เพิ่มเครื่องหมายแยกกลุ่มของตัวอักขระตามพื้นที่ใช้งาน

.3 ให้แสดงทศนิยมสามตำแหน่ง

10.3 ความกว้างของอักขระมีสิบตัวอักษรและเป็นทศนิยมสามตำแหน่ง

ตัวอย่างที่ 3.8-3.13 แสดงตัวอย่างของการใช้งานการจัดรูปแบบการแสดงผลดังกล่าว

ตัวอย่างที่ 3.8 จงจัดรูปแบบของตัวเลขจำนวนเต็ม long=1234567 โดยจัดรูปแบบด้วย %d

รหัสต้นฉบับ :

(1)	public class Ex_3_8{	//เริ่มต้นคลาส
(2)	public static void main(String argv[])	//เมธอดหลัก
(3)	{	//เริ่มคำสั่งของเมธอดหลัก
(4)	long number=1234567;	//ประกาศตัวแปรจำนวนเต็ม
(5)	System.out.format("%d",number);	//แสดงผลตามรูปแบบ %d
(6)	}	//จบส่วนคำสั่งของเมธอดหลัก
(7)	}	//จบคลาส

ผลการรัน แสดงดังนี้

1234567

ตัวอย่างที่ 3.9 จงจัดรูปแบบของตัวเลขจำนวนเต็ม long=1234567 โดยจัดรูปแบบด้วย %012d

รหัสต้นฉบับ :

(1)	public class Ex_3_9 {	//เริ่มต้นคลาส
(2)	public static void main(String argv[])	//เมธอดหลัก
(3)	{	//เริ่มคำสั่งของเมธอดหลัก
(4)	long number=1234567;	//ประกาศตัวแปรจำนวนเต็ม
(5)	System.out.format("%012d",number);	//แสดงผลตามรูปแบบ %012d
(6)	}	//จบส่วนคำสั่งของเมธอดหลัก
(7)	}	//จบคลาส

ผลการรัน แสดงดังนี้

000001234567

ตัวอย่างที่ 3.10 จงจัดรูปแบบของตัวเลขจำนวนเต็ม long=1234567 โดยจัดรูปแบบด้วย %,d

รหัสต้นฉบับ :

(1)	public class Ex_3_10 {	//เริ่มต้นคลาส
(2)	public static void main(String argv[])	//เมธอดหลัก
(3)	{	//เริ่มคำสั่งของเมธอดหลัก
(4)	long number=1234567;	//ประกาศตัวแปรจำนวนเต็ม
(5)	System.out.format("%,d",number);	//แสดงผลตามรูปแบบ %,d
(6)	}	//จบส่วนคำสั่งของเมธอดหลัก
(7)	}	//จบคลาส

ผลการรัน แสดงผลดังนี้

1,234,567

ตัวอย่างที่ 3.11 จงจัดรูปแบบของตัวเลข 12345.67 โดยจัดรูปแบบด้วย %10.3f, %10.2f และ%,10.4f

รหัสต้นฉบับ :

(1)	public class Ex_3_11 {	//เริ่มต้นคลาส
(2)	public static void main(String argv[])	//เมธอดหลัก
(3)	{	//เริ่มคำสั่งของเมธอดหลัก
	//ส่วนการแสดงผลตามรูปแบบของ %f ตามที่โจทย์กำหนด	
(4)	System.out.format("%10.3f%n",12345.67);	
(5)	System.out.format("%10.2f%n",12345.67);	
(6)	System.out.format("%,10.4f",12345.67);	
(7)	}	//จบส่วนคำสั่งของเมธอดหลัก
(8)	}	//จบคลาส

ผลการรัน แสดงดังนี้

12345.670

12345.67

12,345.6700

ตัวอย่างที่ 3.12 จงเขียนโปรแกรมแสดงวันที่ โดย จัดให้อยู่ในรูปแบบ "%tB %te, %tY%n"

รหัสต้นฉบับ :

```
(1) import java.util.Calendar; //ส่วนการนำคลาสปฏิทินเข้ามาใช้งาน
(2) public class Ex_3_12 { //เริ่มต้นคลาส
(3)     public static void main(String argv[]) //เมธอดหลัก
(4)     { //เริ่มคำสั่งของเมธอดหลัก
        //ส่วนการแสดงผลตามที่โจทย์กำหนด
(5)         System.out.format("%tB %te, %tY%n",Calendar.getInstance(),
(6)         Calendar.getInstance(),Calendar.getInstance());
(7)     } //จบส่วนคำสั่งของเมธอดหลัก
(8) } //จบคลาส
```

ผลการรัน แสดงดังนี้

พฤษภาคม 16, 2552

ตัวอย่างที่ 3.13 จงเขียนโปรแกรมแสดงวันที่ของเครื่องแบบสั้น

รหัสต้นฉบับ :

```
(1) import java.util.Calendar; //ส่วนการนำเอาคลาสวันที่มาใช้ร่วม
(2) public class Ex_3_13 { //เริ่มต้นคลาส
(3)     public static void main(String argv[]) //เมธอดหลัก
(4)     { //เริ่มคำสั่งของเมธอดหลัก
        //ส่วนการแสดงผลตามที่โจทย์กำหนด
(5)         System.out.format("%tD%n",Calendar.getInstance());
(6)         System.out.format("%tI:%tM %tp",Calendar.getInstance(),
```

```
(7)     Calendar.getInstance(),Calendar.getInstance());
(8)     }                                     //จบส่วนคำสั่งของเมธอดหลัก
(9)     }                                     //จบคลาส
```

ผลการรัน แสดงดังนี้

05/16/52

10:24 หลังเที่ยง

การใช้งานคำสั่ง System.out.printf

พอล เดทเทล และ ฮาร์เวย์ เดทเทล (Paul Deitel and Harvey Deitel, 2010, p.46) กล่าวว่า System.out.printf เมธอดนี้ f หมายถึง formatted ใช้แสดงผลข้อมูลโดยการกำหนดรูปแบบการแสดงผล คำสั่งนี้ใช้สำหรับจัดรูปแบบการแสดงผล ที่คล้ายกับการใช้ System.out.format และหากผู้เขียนโปรแกรมมีประสบการณ์ในการใช้งานคำสั่งของภาษาซี มาแล้วจะพบว่ารูปแบบการใช้งานคล้ายกันมาก รูปแบบการใช้งานดังนี้

```
System.out.printf("รูปแบบการแสดงผล", ตัวแปรหรือค่าที่จะแสดงผล);
```

สำหรับรูปแบบการแสดงผลที่บังคับด้วย % จะใช้รูปแบบดังที่แสดงไปแล้วในคำสั่ง System.out.printf ตัวอย่างต่อไปนี้แสดงการจัดรูปแบบดังกล่าว

ตัวอย่างที่ 3.14 จงจัดรูปแบบของตัวเลขจำนวนเต็ม long=1234567 โดยจัดรูปแบบด้วย %d %d %+d ด้วย System.out.printf

รหัสต้นฉบับ :

```
(1) public class Ex_3_14 {                                     //เริ่มต้นคลาส
(2)     public static void main(String argv[])               //เมธอดหลัก
(3)     {                                                     //เริ่มคำสั่งของเมธอดหลัก
(4)         long n=123456;                                     //ประกาศตัวแปรจำนวนเต็มแบบยาวและกำหนดค่า
```

```

(5)          System.out.printf("%d %d %+d",n,n,n); //แสดงผลตัวแปรตามรูปแบบที่ต้องการ
(6)      } //จบส่วนคำสั่งของเมธอดหลัก
(7)      } //จบคลาส

```

ผลการรัน แสดงดังนี้

```
123456 123,456 +123,456
```

ตัวอย่างที่ 3.15 จงจัดรูปแบบของตัวเลข 12345.67 โดยจัดรูปแบบด้วย %10.3f, %10.2f และ%,10.4f โดยใช้ System.out.printf

รหัสต้นฉบับ :

```

(1)  public class Ex_3_15 { //เริ่มต้นคลาส
(2)      public static void main(String argv[]) //เมธอดหลัก
(3)      { //เริ่มคำสั่งของเมธอดหลัก
          //ส่วนการแสดงผลตามที่โจทย์กำหนด
(4)          System.out.printf("%10.3f\n",12345.67);
(5)          System.out.printf("%10.2f\n",12345.67);
(6)          System.out.printf("%,10.4f",12345.67);
(7)      } //จบส่วนคำสั่งของเมธอดหลัก
(8)  } //จบคลาส

```

ผลการรันคือ

```
12345.670
12345.67
12,345.6700
```

การจัดรูปแบบเลขทศนิยม

`java.text.DecimalFormat` เป็นแพ็คเกจสำหรับการจัดรูปแบบตัวเลขทศนิยม การจัดการด้วยวิธีนี้ต้องเขียนโปรแกรมเพิ่มมากขึ้น โดยใช้งานผ่านคอนสตรัคเตอร์ของคลาส `DecimalFormat` คลาสนี้อยู่ใน `java.text` ซึ่งการใช้งานจำเป็นต้อง `import java.text.DecimalFormat` เข้ามาร่วมในคลาสด้วยเสมอ รูปแบบการใช้งานดังนี้

```
DecimalFormat ชื่อตัวแปร = new DecimalFormat(รูปแบบที่ต้องการจัด);
```

ตัวอย่างการใช้งาน เช่น

```
DecimalFormat myFormat = new DecimalFormat("###,###.###");
```

ตารางที่ 3.2 แสดงรูปแบบและตัวอย่างการใช้งาน

ตารางที่ 3.2 ตารางการแสดงค่าเบื้องต้น

ค่า	รูปแบบ	ผล	คำอธิบาย
123456.789	###,###.###	123,456.789	แสดงผลและมีเครื่องหมาย , แบ่งตามรูปแบบที่กำหนด
123456.789	###.##	123456.79	แสดงผลโดยใช้การประมาณตามค่าของทศนิยม
123.456	000000.000	000123.780	แสดงผลโดยกำหนดให้ส่วนที่เกินจากตัวเลขที่กำหนดให้เป็นเลขศูนย์
12345.67	\$###,###.###	\$12,345.67	แสดงผลโดยมีเครื่องหมาย , แบ่งตามรูปแบบและมีเครื่องหมาย \$ ด้วย
12345.67	###,###.000	12,345.670	แสดงผลโดยมีเครื่องหมาย , แบ่งตามรูปแบบและมีเลขศูนย์ต่อท้าย

(ที่มา : Lesson: Numbers and Strings -<http://java.sun.com/docs/books/tutorial/java/data/index.html>)

ตัวอย่างที่ 3.16 จงเขียนโปรแกรมจัดรูปแบบ 12345.67 โดยใช้ DecimalFormat โดยจัดรูปแบบให้ดังนี้
 ###,###.## 0000000.00 \$###,###.### และ ###,###.0000

รหัสต้นฉบับ :

```
(1)  import java.text.*;                               //ส่วนการนำเอาคลาส ข้อความมาใช้ร่วม
(2)  public class Ex_3_16 {                             //เริ่มต้นคลาส
(3)      public static void main(String argv[])        //เมธอดหลัก
(4)      {                                             //เริ่มคำสั่งของเมธอดหลัก
        //ส่วนการประกาศตัวแปรวัตถุและกำหนดรูปแบบตามที่โจทย์กำหนด
(5)          DecimalFormat myFormat1 =new DecimalFormat("###,###.##");
(6)          DecimalFormat myFormat2 =new DecimalFormat("0000000.00");
(7)          DecimalFormat myFormat3 =new DecimalFormat("$###,###.###");
(8)          DecimalFormat myFormat4 =new DecimalFormat("###,###.0000");
        //ส่วนการแสดงผลตามรูปแบบที่กำหนดไว้
(9)          System.out.printf(myFormat1.format(12345.67)+"\n");
(10)         System.out.printf(myFormat2.format(12345.67)+"\n");
(11)         System.out.printf(myFormat3.format(12345.67)+"\n");
(12)         System.out.printf(myFormat4.format(12345.67));
(13)     }                                             //จบส่วนคำสั่งของเมธอดหลัก
(14) }                                             //จบคลาส
```

ผลการรัน แสดงดังนี้

```
12,345.67
0012345.67
$12,345.67
12,345.6700
```


สรุป

การเขียนโปรแกรมเพื่อควบคุมการแสดงผลเป็นสิ่งแรกๆ ที่ผู้เริ่มฝึกหัดเขียนโปรแกรมต้องเรียนรู้ และหยังความสามารถของการเรียนรู้ว่าสามารถใช้โปรแกรมภาษานั้นได้หรือไม่ คำสั่งแสดงผลของภาษาจาวาใช้งานผ่านคำสั่ง `System.out` เป็นหลัก สามารถสร้างการแสดงผลได้ทั้งออบเจกต์ ตัวเลข ค่าทางตรรกะ คำสั่งที่เกี่ยวข้องที่สามารถจัดรูปแบบการแสดงผลได้ คือ `System.out.print` เป็นคำสั่งแสดงผลข้อมูลโดยไม่ขึ้นบรรทัดใหม่ คำสั่ง `System.out.println` ส่งแสดงผลแล้วขึ้นบรรทัดใหม่ ส่วนคำสั่ง `System.out.format` และ `System.out.printf` แสดงผลโดยการกำหนดตัวเลขด้วยรูปแบบที่แตกต่างกัน การแสดงผลด้วยคำสั่งต่าง ๆ เหล่านี้สามารถแทรกรหัสหลุดพ้นเพื่อกำกับการแสดงผลข้อมูลได้ นอกจากนี้ยังมีคำสั่งที่มาจากคลาส `java.text.DecimalFormat` ที่สามารถสร้างรูปแบบการแสดงผลได้อีกด้วย

แบบฝึกหัดท้ายบท

ตอนที่ 1 จงเขียนเครื่องหมาย ✓ หน้าข้อที่กล่าวถูกต้อง และ เขียนเครื่องหมาย ✗ หน้าข้อที่กล่าวผิด

1. _____ การแสดงผลด้วย System.out.print จะไม่ขึ้นบรรทัดใหม่
2. _____ การแสดงผลด้วย System.out.print ไม่สามารถจัดรูปแบบได้
3. _____ System.out.print ('\n'); สั่งให้ขึ้นบรรทัดใหม่ได้
4. _____ System.out.print ('\') จะแสดง ' เมื่อรันโปรแกรม
5. _____ System.out.println(“%d”, 1234.6); ไม่สามารถแสดงผลได้
6. _____ การแสดงผลของภาษาจาวาแสดงได้เฉพาะข้อความเท่านั้น
7. _____ การแสดงผลโดยการสร้างรูปแบบเองต้อง import java.text เข้าร่วมด้วย
8. _____ การแสดงผลของภาษาจาวาไม่สามารถแสดงผลในรูปแบบกราฟิกได้
9. _____ การแสดงผลแบบวันที่ เมื่อไปรันยังต่างประเทศจะได้รูปแบบวันที่แตกต่างกัน
10. _____ ข้อมูล boolean เมื่อแสดงผลจะแสดงเป็น 0 และ 1

ตอนที่ 2 จงตอบคำถามต่อไปนี้

1. จงออกและเขียน โปรแกรมให้มีลักษณะดังนี้

This is program Test

=====

My name is Praram

=====

F1-Help F2-Save F3-Edit F4-Tutorial F10-Exit

=====

2. จงเขียนโปรแกรมแสดงข้อมูลส่วนตัว ได้แก่ ชื่อ สกุล ที่อยู่ และความใฝ่ฝัน
3. จงเขียนโปรแกรมแสดงสูตรคูณแม่ 2
4. กำหนดให้ x=10, y=20, z=30 จงเขียนโปรแกรมแสดงตัวเลขดังกล่าวให้ห่างกันด้วยแท็บ
5. กำหนด 987654 จงเขียน โปรแกรมจัดรูปแบบตัวเลขดังกล่าวให้แสดงแตกต่างกันอย่างน้อย 4 รูปแบบ
6. กำหนด char a='x', char b='y', char c='z' จงเขียน โปรแกรมให้แสดงตัวอักษรดังกล่าวใน

รูปแบบของเลขฐานสิบหก

7. กำหนดตัวเลข `int x=10`; จงเขียนโปรแกรมเพื่อแสดงค่าในเลขฐานสอง
8. จงเขียนโปรแกรมให้แสดง ข้อความดังนี้ “เราคือผู้กล้า ‘แห่งบางปะกง’ ”
9. จงเขียนโปรแกรมสร้างตารางเพื่อแสดงรายรับรายจ่ายของตนเองช่วงหนึ่งสัปดาห์
10. จงเขียนโปรแกรมแสดงจำนวนเต็มและชนิดจำนวนจริงชนิดละ 3 ค่าแล้วแสดงผลให้มีรูปแบบดังนี้

ตัวเลขจำนวนเต็มที่ 1	ตัวเลขจำนวนจริงตัวที่ 3(ทศนิยม 2 แห่ง)
----------------------	--

ตัวเลขจำนวนเต็มที่ 3	ตัวเลขจำนวนจริงตัวที่ 1(ทศนิยม 4 แห่ง)
----------------------	--

ตัวเลขจำนวนเต็มที่ 2	ตัวเลขจำนวนจริงตัวที่ 2(ทศนิยม 3 แห่ง)
----------------------	--

เอกสารอ้างอิง

- กิตติ ภัคดีวัฒนกุล. (2543 ก). **JAVA ฉบับพื้นฐาน**. กรุงเทพฯ: ไทยเจริญการพิมพ์.
- ชญัญชัย ตริภาก. (2553). **คอมพิวเตอร์และการเขียนโปรแกรม (Computers and Programming) C และ Java**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- ธีรวัฒน์ ประกอบผล. (2545). **การโปรแกรมภาษาซีสำหรับงานวิทยาศาสตร์**. กรุงเทพฯ: ศ.ศ.ท.
- _____. (2552). **คู่มือการเขียนโปรแกรมภาษา Java**. กรุงเทพฯ: ชัคเซส มีเดีย.
- _____. (2553). **คู่มือการเขียนโปรแกรมภาษา Java**. กรุงเทพฯ: ซิมพลิฟลาย.
- พนิดา พานิชกุล. (2548). **Object-Oriented ฉบับพื้นฐาน**. กรุงเทพฯ: เคทีพี แอนด์ คอนซัลท์.
- _____. (2554). **การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษาจาวา**. พิมพ์ครั้งที่ 5. กรุงเทพฯ: เคทีพี แอนด์ คอนซัลท์.
- พิเชษฐ ศิริรัตนไพศาลกุล. (2553). **หลักการเขียนโปรแกรมเบื้องต้นด้วยภาษา JAVA**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- อนรรฆมนงค์ คุณมณี. (2551). **คู่มือเขียนโปรแกรมภาษา JAVA ฉบับผู้เริ่มต้น**. กรุงเทพฯ: ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์.
- David J. Eck. (2011). **Introduction to Programming Using Java**. NY: Department of Mathematics and Computer Science Hobart and William Smith Colleges Geneva.
- Harvey M. Deitel. (2002). **Java How to Program**. 4th Edition Reviewers. USA: Prentice Hall.
- _____. (2004). **Java How to Program**. 6th Edition. USA: Prentice Hall.
- _____. (2006). **Java How to Program**. 7th Edition. USA: Prentice Hall.
- Paul Deitel and Harvey Deitel. (2010). **Java How to Program**. 8th Edition. USA: Pearson Education.
- Richard F. Raposa. (2003). **Java in 60 Minutes a Day**. USA: Wiley Publishing, Inc., Indianapolis, Indiana.
- Sharon Zakhour, Scott Hommel, Jacob Royal, Isaac Rabinovitch, Tom Risser, and Mark Hoeber. (2006). **The Java Tutorial**. 4th Edition: A Short Course on the Basics. USA: Addison Wesley Professional.

Stuart Reges and Marty Stepp. (2011). **Building Java Programs A Back to Basics Approach**. USA:
Pearson Education, Inc., publishing as Addison-Wesley.

บทที่ 10

เมธอดและการเรียกใช้คลาส

การเขียนโปรแกรมโดยนำคำสั่งต่าง ๆ มาไว้ในโปรแกรมเดียวกันทั้งหมดไม่เหมาะสำหรับโปรแกรมที่มีความซับซ้อนมาก ดังนั้นจึงมีวิธีการที่แยกความซับซ้อนให้น้อยลงโดยการแบ่งโปรแกรมให้ย่อยลง ซึ่งต้องออกแบบให้แต่ละส่วนที่ย่อยลงแล้วทำหน้าที่เฉพาะอย่าง เรียกว่าเมธอด (Method) โดยเมธอดจะมีหลากหลายรูปแบบ บทนี้จะนำเสนอรายละเอียดของเมธอดแบบต่าง ๆ ที่มีประโยชน์ และความเหมาะสมกับการใช้งานที่แตกต่างกัน นอกจากนั้นแล้วสามารถที่จะแบ่งการทำงานออกเป็นคลาส ที่ทำหน้าที่แตกต่างกันเพื่อให้โปรแกรมได้ทำงานแยกส่วนกันได้อีกด้วย ซึ่งในภาษาจาวาถือว่าแต่ละโปรแกรมคือคลาสนั่นเอง เนื้อหาในบทนี้จะกล่าวถึงเมธอดและการสร้างคลาสแยกเพื่อทำงานประสานกันเพื่อใช้งานสำหรับแก้ปัญหาใดปัญหาหนึ่งตามที่คุณเขียนโปรแกรมต้องการ

ความหมายและความสำคัญของเมธอด

พนิดา พานิชกุล (2554, หน้า 214) กล่าวว่า เมธอดคือระเบียบวิธีในการทำงานของงานใดงานหนึ่งซึ่งประกอบด้วยขั้นตอนต่าง ๆ มากมายทำงานนั้นบรรลุผลได้ขั้นตอนต่าง ๆ เหล่านั้นก็เปรียบเสมือนกับคำสั่งต่าง ๆ ของภาษาจาวาที่นำมาประกอบรวมกันเข้าเพื่อให้ 1 เมธอดสามารถทำงาน 1 อย่าง

พิเชษฐ ศิริรัตนไพศาลกุล (2553, หน้า 140) กล่าวว่า เมธอด เป็นกระบวนการในการประมวลผลหรือส่งให้โปรแกรมทำงานต่าง ๆ ตามหน้าที่ที่เราสั่งไว้ในเมธอด ซึ่งย่อมเกี่ยวข้องกับตัวแปรที่อยู่ในออบเจกต์นั้น ๆ เมธอดถูกแบ่งออกเป็น 2 ชนิดคือ

- คลาสเมธอด (Class Method) คือ เมธอดที่สามารถประมวลผลได้เองโดยไม่จำเป็นต้องสร้างออบเจกต์ใด ๆ ขึ้นมาใช้เลย (เหมือนเมธอด main()) ไม่ต้องสร้างออบเจกต์ใด ๆ มาเรียกใช้เลย การประกาศคล้ายเมธอดก็ทำได้เช่นเดียวกันกับคลาสสวาริเบิ้ลคือต้องใช้คำสั่ง static นำหน้าเสมอ

- อินสแตนซ์เมธอด (Instance Method) คือ เมธอดที่ต้องสร้างออบเจกต์ขึ้นมาก่อนถึงจะเรียกใช้เมธอดนั้นได้ การประกาศก็ทำได้ปกติทั่วไปไม่ต้องคำนึงถึงคำว่า static นำหน้า

ดังนั้น **เมธอด** หมายถึง โมดูลย่อยหรือโปรแกรมย่อยที่ทำหน้าที่อย่างใดอย่างหนึ่งเพื่อสนับสนุนโปรแกรมหลักที่เขียนขึ้น วัตถุประสงค์ของการสร้างเมธอดเพื่อแบ่งงานโปรแกรมที่มีขนาดใหญ่ให้มีกลุ่มโปรแกรมย่อย ๆ ลงมา เมธอดแบ่งตามการเรียกใช้งานได้ 3 ประเภท คือ

1. เมธอดที่ไม่มีการรับค่าใด ๆ ในการเรียกใช้เมธอด
2. เมธอดที่ต้องรับค่าพารามิเตอร์ (Parameter) เข้าไปทำงานในเมธอด
3. เมธอดที่ต้องรับค่าพารามิเตอร์เข้ามาในการทำงานและส่งค่ากลับไปให้ตำแหน่งที่เรียกใช้

เมธอดมีประโยชน์หลายประการ เช่น

1. ทำให้ไม่ต้องเขียนคำสั่งเดิมซ้ำกันหลาย ๆ ครั้ง เพราะหากเป็นกลุ่มคำสั่งที่ยาว ๆ ในโปรแกรมยิ่งจะทำให้โปรแกรมมีขนาดใหญ่ขึ้น เมธอดจะสร้างกลุ่มคำสั่งไว้เพียงชุดเดียวแล้วเรียกใช้โดยไม่ต้องเขียนกลุ่มคำสั่งนั้นใหม่ซ้ำ ๆ อีก
2. ทำให้ตรวจหาที่ผิดของโปรแกรมง่ายขึ้นเนื่องจากการจัดการโดยใช้เมธอดจะแบ่งงานออกเป็นส่วน ๆ เราทราบว่าแต่ละส่วนทำหน้าที่อย่างไร โดยเฉพาะ ดังนั้นจึงเป็นการง่ายในการที่จะหาที่ผิดของโปรแกรมว่ามาจากงานส่วนใด
3. ทำให้โปรแกรมมีขนาดเล็กลง เพราะไม่ต้องเขียนคำสั่งซ้ำ ๆ กันหลาย ๆ ครั้ง ตลอดจนสามารถแบ่งเก็บเอาไว้ในโปรแกรมย่อยอื่น ๆ ได้อีก จึงทำให้โปรแกรมมีขนาดเล็กลง
4. ทำให้การเรียกใช้เมธอดไปใช้งานได้กับโปรแกรมอื่น ๆ ได้อีก เป็นการช่วยสร้างงานขึ้นมาครั้งเดียวแล้วสามารถเรียกใช้งานได้หลาย ๆ ครั้งอย่างคุ้มค่า
5. ทำให้สามารถแยกงานกันทำได้ หากเป็นระบบงานขนาดใหญ่จะสามารถทำงานได้อย่างอิสระ

พินดา พานิชกุล (2554, หน้า 215) กล่าวว่า โครงสร้างทั่วไปของเมธอด (ยกเว้น main()) มีรูปแบบดังนี้

```
Visibility return_data_type methodName(Parameter_list)
{...คำสั่งอื่น ๆ ภายใน Method ... }
```

โดยที่ Visibility หรือ Accessibility ได้แก่

- public คือการสร้างเมธอดเพื่อสามารถนำไปใช้ได้กับทุกคลาส
- private คือ การสร้างเมธอดเพื่อใช้งานภายในคลาสที่มีเมธอดนั้นอยู่เท่านั้นไม่สามารถนำไปใช้ในคลาสอื่น ๆ ได้
- protected คือการสร้างเมธอดเพื่อให้สามารถนำไปใช้ในคลาสอื่น ๆ ได้เมื่อคลาสนั้นขยายมาจากคลาสดังกล่าว

- default คือ ไม่ได้กำหนดค่า Accessibility ไว้หมายถึง สร้างเมธอดเพื่อสามารถเรียกใช้งานได้จากคลาสอื่น ๆ ภายในแพ็คเกจเดียวกัน

return_data_type ได้แก่การกำหนดชนิดของผลลัพธ์ที่เมธอดนั้นจะส่งค่ากลับใน 1 เมธอด จะต้องส่งผลลัพธ์อย่างน้อย 1 ค่ากลับไปยังคำสั่งที่เรียกมา (ยกเว้นคีย์เวิร์ด เป็น void) ทั้งนี้ return_data_type อาจเป็นชนิดทั่วไป (เช่น integer, character) หรือเป็นชนิดซับซ้อนเช่น Array หรือ Object ก็ได้

Parameter_list เป็นชื่อของตัวแปรที่รับข้อมูลที่ส่งมาจากคำสั่งอื่นที่มาเรียกใช้งาน ถ้าไม่มีการส่งค่ามาให้ก็ไม่จำเป็นต้องมีก็ได้ กรณีที่มี parameter_list มากกว่าหนึ่งตัวให้คั่นด้วยเครื่องหมายคอมมา (Comma) (,) ตัวแปรที่ระบุแต่ละตัวจะต้องระบุชนิดตัวแปรนำหน้าเสมอ

นอกจากนั้นธีรวัฒน์ ประกอบผล (2552, หน้า 124) ได้กล่าวว่า โปรแกรมทุกโปรแกรมจะต้องมีเมธอดที่ชื่อว่า main() ซึ่งถือว่าเป็นเมธอดหลักของโปรแกรม ในเมธอดหนึ่ง ๆ จะประกอบด้วยสองส่วนใหญ่ ๆ คือ ส่วนหัว (Header) และส่วนสเททเมนต์ หรือเรียกว่า Body ซึ่งเก็บการทำงานต่าง ๆ เอาไว้ ดังเมธอด main() ต่อไปนี้

```
Header-----> public static void main(String args[]){
Body----->   System.out.println("Hello World");
                }
```

รูปแบบของเมธอดเป็นดังนี้

```
Modifier Return_Type MethodName(Parameter_list)
{
    ชุดคำสั่งหรือสเททเมนต์ต่าง
return (ตัวแปรหรือข้อมูล);
}
```

โดยที่ Modifier เป็นส่วนที่บอกว่า เมธอดนี้สามารถใช้ได้ในระดับใดในการเขียนโปรแกรมแบบพื้นฐานแล้วส่วนนี้ประกอบด้วย public static

Return_Type ส่วนนี้จะบอกว่าเมื่อเมธอดนี้ถูกเรียกใช้แล้วจะมีการส่งค่ากลับหรือไม่ถ้ามีจะส่งกลับเป็นข้อมูลประเภทใด ถ้าหากเมธอดไม่มีการส่งค่ากลับจะนำหน้าด้วย void

MethodName เป็นชื่อของเมธอดถ้าหากส่วนใดต้องการใช้เมธอดนี้ก็จะเรียกชื่อเมธอดได้เลย สำหรับการตั้งชื่อให้เป็นไปตามกฎการตั้งชื่อตัวแปร

Parameter_list เป็นตัวแปรที่ส่งข้อมูลเข้าไปในเมธอด ถ้าหากมีตัวแปรหลายตัวจะใช้เครื่องหมายคอมมา (,) คั่นระหว่างตัวแปรถ้าไม่มีการส่งตัวแปรว่าใช้วงเล็บว่าง

Return เป็นส่วนที่ใช้ส่งค่ากลับให้กับเมธอด ถ้าเมธอดใดไม่มีการส่งค่ากลับก็จะไม่ใช่ส่วนนี้ สำหรับตัวแปรที่ส่งค่ากลับจะต้องเป็นประเภทเดียวกันกับที่ประกาศไว้ในส่วนหัวของเมธอด

จากที่กล่าวมาสรุปได้ว่า รูปแบบของเมธอดที่ใช้กันทั่วไปมีดังนี้

```
ส่วนหัว-----> Modifier Return_Type MethodName(Parameter_list)
                { // เริ่มส่วนตัวเมธอด
ส่วนตัว-----> //เขียนรหัสต้นฉบับ
ส่วนการคืนค่า-> [return Value_Type]
                }// จบส่วนตัวเมธอด
```

โดยที่ส่วนหัวจะประกอบไปด้วยส่วนของการระบุค่าการเข้าถึงเมธอด เช่น public, private, protected และอาจตามด้วย static เช่น public static เป็นต้น

ส่วนของ Return_Type เป็นการบอกว่าเมื่อเมธอดนี้ทำงานแล้วจะคืนค่าใดออกไป ซึ่งอาจเป็น int, float หรือข้อมูลอ้างอิง เช่น อาร์เรย์หรือวัตถุของคลาสก็ได้

ส่วนของ MethodName คือ กำหนดชื่อของเมธอด โดยเป็นไปตามกฎการตั้งชื่อของภาษา

ส่วนของ Parameter_list คือ ส่วนการระบุพารามิเตอร์ที่ต้องใส่ให้กับเมธอดเมื่อมีการเรียกใช้งาน

ส่วนตัวเมธอด ซึ่งเป็นส่วนการเขียนคำสั่งต่าง ๆ

ส่วนการคืนค่า ส่วนนี้จะต้องคืนเป็นค่าตัวแปรหรือค่าที่ระบุโดยตรงก็ได้ โดยที่ต้องตรงกับค่าที่ระบุเอาไว้ในส่วนหัว

ตัวอย่างเมธอด

```
public static int Test(int x, int y)
{
    int z=x+y;
    return z;
}
```

อธิบายได้ว่า public static เป็นส่วนของ Modifier

int เป็น Return_Type

Test เป็น MethodName

int x และ int y เป็น Parameter_list

int z=x+y เป็นส่วนตัวของเมธอด
return z เป็นการกำหนดค่าที่คืนกลับไปเมื่อถูกเรียกใช้งาน

ตำแหน่งของการวางเมธอดในโปรแกรม

โดยทั่วไปแล้วโครงสร้างของแต่ละโปรแกรมจะมีส่วนประกอบ ดังนี้

- 1) ส่วนหัว โปรแกรมที่เรียก import เพื่อเรียกใช้เมธอดจากคลาสไลบรารีของภาษาจาวา
- 2) ส่วนการประกาศตัวแปรสาธารณะ (Global variables หรือ Attribute)
- 3) ส่วนเมธอดหรือส่วนตัวของเมธอด (Body method) ซึ่งหมายถึงการวางเมธอดไว้ในโปรแกรมนั้น ซึ่งส่วนนี้เป็นส่วนที่เป็นเนื้อหาหลักที่จะกล่าวถึงในบทนี้
- 4) ส่วนของโปรแกรมหลัก ซึ่งก็คือเมธอดหลัก (main method)

โครงสร้างดังกล่าวนี้ แสดงได้ดังภาพที่ 10.1

1. หัวโปรแกรม ส่วน import และ ระบุชื่อคลาส
2. ประกาศตัวแปรสาธารณะ
3. ตัวเมธอด
4. ตัวเมธอดหลัก (main)
3. ตัวเมธอด

ภาพที่ 10.1 แสดงรายละเอียดของส่วนประกอบโปรแกรม

โปรแกรมหนึ่ง ๆ จะมีส่วนประกอบเต็มรูปแบบ ดังภาพด้านบน ตัวอย่างโปรแกรมต่อไปนี้เป็นการเขียนโปรแกรมให้มีลักษณะโครงสร้างอิงรูปภาพที่ 10.1 โดยโปรแกรมนี้จะแสดงข้อความ "My name is Praram" บนจอภาพ 3 ครั้ง

```
import java.io.*;
import java.text.* ----- 1
public class Example{
int x;
int y; ----- 2
public void show1(){
```

```

System.out.print("My name is Praram\n"); -----3
}
public static void main(String args[]){ -----4
    Example ex=new Example();
    ex.show1(); ex.show1(); -----4
    ex.show2();
}
public void show2(){
    System.out.print("My name is Praram\n"); }-----3
}

```

เมธอดแบบไม่คืนค่า

เมธอดที่ไม่มีการคืนค่า จะเป็นเมธอดที่เขียนขึ้นต้นด้วย void โดยจะมีหลายรูปแบบ ทั้งที่มีการผ่านค่าพารามิเตอร์และไม่มีการผ่านค่าพารามิเตอร์ ตำแหน่งการวางเมธอดในโปรแกรมจะเหมือนกับหัวข้อที่อธิบายมาแล้ว

ลักษณะของเมธอดประเภทนี้จะมี void นำหน้าเมธอด เสมอ และในการเรียกใช้งานก็สามารถเรียกผ่านชื่อเมธอดได้ทันที

- การประกาศใช้งานหรือการสร้างเมธอด

การประกาศตอนสร้างเมธอดจะต้องระบุ void ไว้หน้าชื่อของเมธอดเสมอ เช่น

```

public static void main(){
    ...
}

```

หมายถึง เมธอด main ไม่มีการคืนค่า เพราะมี void กำกับไว้ข้างหน้า

```

public void Test(int x){
    ...
}

```

หมายถึงเมธอด Test ไม่มีการคืนค่า เพราะมี void กำกับไว้ข้างหน้า

ตัวอย่างที่ 10.1 จงเขียนโปรแกรมสร้างเมธอดแสดงข้อความ I Love Java. และให้เมธอดหลักเรียกใช้เมธอดดังกล่าว จำนวน 10 ครั้ง

รหัสต้นฉบับ :

```
(1) public class Ex_10_1 { //เริ่มต้นคลาส
(2)     public static void main(String args[]){ //ส่วนของเมธอดหลัก
(3)         Ex_10_1 obj = new Ex_10_1(); //ประกาศออบเจกต์ของตัวเอง
           //เรียกใช้เมธอดแสดงผล 10 ครั้ง
(4)         obj.showMessage();
(5)         obj.showMessage();
(6)         obj.showMessage();
(7)         obj.showMessage();
(8)         obj.showMessage();
(9)         obj.showMessage();
(10)        obj.showMessage();
(11)        obj.showMessage();
(12)        obj.showMessage();
(13)        obj.showMessage();
(14)     } //จบเมธอดหลัก
(15)     public void showMessage(){ //เริ่มส่วนของตัวเมธอดแสดงผล
(16)         System.out.println("I Love Java."); //แสดงข้อความตามที่โจทย์กำหนด
(17)     } //จบเมธอดแสดงผล
(18) }
```

การทำงานของโปรแกรม จะเริ่มรันที่ส่วนเมธอดหลักก่อน หลังจากนั้นโปรแกรมหลักจะเรียกใช้เมธอด showMessage() ตามจำนวน 10 ครั้ง ผลการรันจะแสดงข้อความ I Love Java จำนวน 10 บรรทัด

ตัวอย่างที่ 10.2 จงเขียนโปรแกรมรับค่าสตริงและจำนวนครั้งของการแสดงผล โดยกำหนดให้ การรับทั้งสองครั้งอยู่ในเมธอด inputData แล้วแสดง โดยใช้ showData ผลตามที่ป้อน

รหัสต้นฉบับ :

```
(1) import java.io.*;                //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_2 {          //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3)   InputStreamReader reader = new InputStreamReader(System.in);
(4)   BufferedReader stdin = new BufferedReader(reader);
(5)   String str1;                  //ประกาศตัวแปรสตริงรับข้อมูล
(6)   int number;                   //ประกาศตัวแปรจำนวนเต็มเพื่อใช้คำนวณ
(7)   public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(8)   {
(9)       Ex_10_2 obj = new Ex_10_2(); //ประกาศออบเจกต์ตัวเอง
(10)      obj.inputData();             //เรียกใช้เมธอดรับข้อมูล
(11)      obj.showData();             //เรียกใช้เมธอดแสดงผล
(12)  }                                //จบเมธอดหลัก
(13)  public void inputData()throws IOException{ //เริ่มต้นเมธอดรับข้อมูล
(14)      System.out.println("Input String :"); //แสดงข้อความให้ผู้ใช้เห็น
(15)      str1=stdin.readLine();       //รับข้อมูลข้อความที่จะแสดง
(16)      System.out.println("Input Times :"); //แสดงข้อความให้ผู้ใช้เห็น
(17)      number=Integer.parseInt(stdin.readLine());//รับข้อมูลจำนวนครั้งที่แสดง
(18)  }                                //จบเมธอดรับข้อมูล
(19)  public void showData(){         //เริ่มเมธอดแสดงผล
(20)      for(int i=1;i<=number;i++)   //กำหนดลูปแสดงจาก i=1 ถึง จำนวนที่รับค่ามา
(21)          System.out.println(str1); //แสดงข้อความที่รับเข้ามา
(22)  }                                //จบเมธอดแสดงผล
(23) }
```

ผลการรันจะให้ผู้ใช้ป้อนสตริง และจำนวนครั้งของการแสดง หลังจากนั้นจะแสดงสตริงที่ป้อนเข้าไป เท่ากับจำนวนครั้งที่ป้อน

Input String :This is the test of Method.

Input Times :20

This is the test of Method.

This is the test of Method.

This is the test of Method.

...

ตัวอย่างที่ 10.3 จากตัวอย่างที่ 10.2 จงเขียนโปรแกรมสร้างให้มีเมธอด จำนวน 3 เมธอด คือ `inputString` ใช้ในการรับค่าสตริงและ `inputNumber` จำนวนครั้ง และเมธอด `myShow` ใช้ในการแสดงผล โดยประกาศตัวแปร `str` และ `number` เป็นตัวแปรสาธารณะ

รหัสต้นฉบับ :

```
(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_3 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3) InputStreamReader reader = new InputStreamReader(System.in);
(4) BufferedReader stdin = new BufferedReader(reader);
(5) String str1; //ประกาศตัวแปรสตริงรับข้อมูล
(6) int number; //ประกาศตัวแปรจำนวนเต็มเพื่อใช้คำนวณ
(7) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(8) { //เริ่มต้นคำสั่งของเมธอดหลัก
(9) Ex_10_3 obj = new Ex_10_3(); //ประกาศออบเจกต์ตัวเอง
(10) obj.inputString(); //เรียกใช้เมธอดรับสตริง
(11) obj.inputNumber(); //เรียกใช้เมธอดรับจำนวนครั้งที่แสดง
(12) obj.showData(); //เรียกใช้เมธอดแสดงผล
(13) } //จบเมธอดหลัก
(14) public void inputString()throws IOException{ //เริ่มต้นเมธอดรับข้อมูล
(15) System.out.println("Input String :"); //แสดงข้อความให้ผู้ใช้น้เห็น
(16) str1=stdin.readLine(); //รับข้อมูลข้อความที่จะแสดง
(17) } //จบเมธอดรับสตริง
(18) public void inputNumber()throws IOException{ //เริ่มเมธอดรับจำนวนครั้ง
```

```

(19) System.out.println("Input Times :"); //แสดงข้อความให้ผู้ใช้เห็น
(20) number=Integer.parseInt(stdin.readLine()); //รับข้อมูลจำนวนครั้งที่แสดง
(21) } //จบเมธอดรับจำนวนครั้ง
(22) public void showData(){ //เริ่มต้นเมธอดแสดงผล
(23) for(int i=1;i<=number;i++) //กำหนดลูปแสดงจาก i=1 ถึง จำนวนที่รับค่ามา
(24) System.out.println(str1); //แสดงข้อความที่รับเข้ามา
(25) } //จบเมธอดแสดงผล
(26) } //จบคลาส

```

ผลการรันจะแสดงว่าเมธอดใดกำลังทำงาน โดยจะให้รับค่าการป้อนค่า สตริง และจำนวนที่ต้องการแสดง หลังจากนั้นจะแสดงสตริงดังกล่าวเท่ากับจำนวนที่ป้อนเข้าไป

Input String :Show Message of Example 3.

Input Times :10

Show Message of Example 3.

Show Message of Example 3.

...

ตัวอย่างที่ 10.4 จงเขียน โปรแกรมรับค่ารัศมีของวงกลมจากผู้ใช้แล้วคำนวณหาพื้นที่แล้วแสดงผลทางจอภาพโดยใช้เมธอด

รหัสต้นฉบับ :

```

(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_4 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3) InputStreamReader reader = new InputStreamReader(System.in);
(4) BufferedReader stdin = new BufferedReader(reader);
(5) String str1; //ประกาศตัวแปรสตริงรับข้อมูล
(6) int number; //ประกาศตัวแปรจำนวนเต็มเพื่อใช้คำนวณ
(7) double area; //ประกาศตัวแปรเพื่อเก็บค่าผลการคำนวณ
(8) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(9) { //เริ่มต้นคำสั่งของเมธอดหลัก

```

```

(10)   Ex_10_4 obj = new Ex_10_4();           //ประกาศออบเจกต์ตัวเอง
(11)   obj.inputData();                       //เรียกใช้เมธอดรับข้อมูล
(12)   obj.calculate();                       //เรียกใช้เมธอดคำนวณ
(13)   obj.showData();                       //เรียกใช้เมธอดแสดงผล
(14)   }                                     //จบเมธอดหลัก
(15)   public void inputData()throws IOException{ //เริ่มต้นเมธอดรับข้อมูล
(16)       System.out.println("Input Radius :"); //แสดงข้อความให้ผู้ใช้เห็น
(17)       number=Integer.parseInt(stdin.readLine()); //รับข้อมูลรัศมีของวงกลม
(18)   }                                     //จบเมธอดรับข้อมูล
(19)   public void calculate(){               //เริ่มต้นเมธอดคำนวณ
(20)       area=Math.PI*(number*number);     //คำนวณเก็บผลไว้ที่ area
(21)   }                                     //จบเมธอดคำนวณ
(22)   public void showData(){               //เริ่มต้นเมธอดแสดงผล
(23)       System.out.format("The area is %.2f",area); //แสดงผลการรคำนวณ
(24)   }                                     //จบเมธอดแสดงผล
(25) }                                     //จบคลาส

```

ผลการรัน จะให้ป้อนรัศมี และแสดงผลการคำนวณ ตัวอย่างดังนี้

```

Input Radius :100
The area is 31415.93

```

โปรแกรมนี้จะให้ผู้ใช้ป้อนค่ารัศมีของวงกลม หลังจากนั้นโปรแกรมจะทำการคำนวณและแสดงผลออกมาทางจอภาพให้ผู้ใช้ได้เห็นและให้สังเกตการประกาศตัวแปรสาธารณะที่มีการเรียกใช้ได้ทั่วไปในโปรแกรม และการตั้งชื่อ ตลอดจนการเขียนคำสั่งในเมธอดนั้น กล่าวคือ เมธอดเหมือนกับโปรแกรมหนึ่งทำงานโดยเฉพาะอย่างใดอย่างหนึ่ง เช่น inputData() ทำหน้าที่รับข้อมูล Calculate() ทำหน้าที่ในการคำนวณ showData() ทำหน้าที่ ในการแสดงผลลัพธ์เป็นต้น

ตัวอย่างที่ 10.5 จงเขียนโปรแกรมรับชื่อและคะแนนของนักศึกษาจำนวน 5 คน แล้วคำนวณหาคะแนนเฉลี่ยโดยใช้เมธอด

รหัสต้นฉบับ :

```
(1) import java.io.*; //ส่วนการนำแพ็คเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_5 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3) InputStreamReader reader = new InputStreamReader(System.in);
(4) BufferedReader stdin = new BufferedReader(reader);
(5) String str[]=new String[5]; //ประกาศอาร์เรย์เพื่อรับค่าตามที่โจทย์ต้องการ
(6) double average; //ประกาศตัวแปรเพื่อเก็บผลการคำนวณ
(7) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(8) { //เริ่มต้นคำสั่งของเมธอดหลัก
(9) Ex_10_5 obj = new Ex_10_5(); //ประกาศออบเจกต์ตัวเอง
(10) obj.inputData(); //เรียกใช้เมธอดรับข้อมูล
(11) obj.calculateAverage(); //เรียกใช้เมธอดคำนวณ
(12) obj.showData(); //เรียกใช้เมธอดแสดงผล
(13) } //จบเมธอดหลัก
(14) public void inputData()throws IOException { //เริ่มต้นเมธอดรับข้อมูล
(15) for(int i=0;i<5;i++){ //วนลูปรับข้อมูลตามที่โจทย์ต้องการ
(16) System.out.println("Input Score of Person "+(i+1)+" :"); //แสดงข้อความให้ผู้ใช้เห็น
(17) str[i]=stdin.readLine(); //รับข้อมูลที่ละรายการ
(18) } //จบลูปรับข้อมูล
(19) } //จบเมธอดรับข้อมูล
(20) public void calculateAverage(){ //เริ่มต้นเมธอดคำนวณ
(21) int sum=Integer.parseInt(str[0])+Integer.parseInt(str[1])+Integer.parseInt(str[2])+
    Integer.parseInt(str[3])+Integer.parseInt(str[4]); //คำนวณผลรวมเก็บค่าใน sum
(22) average=((double)sum)/5; //คำนวณค่าเฉลี่ยเก็บใน average
(23) } //จบเมธอดคำนวณ
(24) public void showData(){ //เริ่มต้นเมธอดแสดงผล
(25) System.out.println("The average is "+average); //แสดงค่าเฉลี่ย
```

(26) }	//จบเมธอดแสดงผล
(27) }	//จบคลาส

ผลการรัน จะให้ผู้ใช้ป้อนคะแนนของนักศึกษาจนครบ แล้วแสดงค่าเฉลี่ย ออกมาดังตัวอย่างผลการรันด้านล่างนี้

```
Input Score of Person 1 :50
Input Score of Person 2 :60
Input Score of Person 3 :70
Input Score of Person 4 :80
Input Score of Person 5 :90
The average is 70.0
```

เมธอดไม่คืนค่าแบบเรียกใช้โดยผ่านค่าพารามิเตอร์

พารามิเตอร์ คือ สิ่งที่ต้องจัดเตรียมที่จะทำตามทีคลาสิกกำหนดไว้เมื่อมีการเรียกใช้เมธอดพารามิเตอร์ในความรู้สึกแล้วจะหมายถึงตัวแปรที่มันจะบรรจุค่าและมีชนิดอาร์กิวเมนต์ (Argument) : ค่าซึ่งต้องจัดเตรียมเมื่อมีการเรียกใช้เมธอด ค่านี้ต้องเป็นค่าเดียวกันกับค่าของพารามิเตอร์ที่เกี่ยวข้อง (Allen B. Downey, 2011, p. 32)

พารามิเตอร์คือค่าที่ต้องผ่านให้กับเมธอดเมื่อมีการเรียกใช้ เช่น `System.out.println("Test");` ซึ่งลักษณะของการสร้างเมธอดเราสามารถที่จะสร้างเมธอดที่เป็นลักษณะดังกล่าวได้ ขอให้ศึกษาจากตัวอย่างต่อไปนี้

การใช้งานเมธอดแบบไม่คืนค่าจะยืดหยุ่นมากขึ้นหากใช้การผ่านค่าพารามิเตอร์เข้ามาช่วย เนื่องจาก การเรียกใช้เมธอดหากไม่มีการผ่านค่าพารามิเตอร์จะทำให้ต้องประกาศตัวแปรสาธารณะมาใช้ ซึ่งไม่นิยมใช้ลักษณะดังกล่าวนี้นัก จึงนิยมใช้การผ่านพารามิเตอร์มากกว่า

ตัวอย่างที่ 10.6 จงเขียนโปรแกรมแสดงค่าเลขยกกำลังสองของค่าที่ผู้ใช้ป้อน

รหัสต้นฉบับ :

(1) <code>import java.io.*;</code>	//ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) <code>public class Ex_10_6 {</code>	//เริ่มต้นคลาส
	//ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3) <code>InputStreamReader reader = new InputStreamReader(System.in);</code>	

```

(4) BufferedReader stdin = new BufferedReader(reader);
(5) static int number; //ประกาศตัวแปรเพื่อเก็บตัวเลขที่จำค่านวน
(6) long square; //ประกาศตัวแปรเพื่อเก็บผลการค่านวน
(7) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(8) { //เริ่มต้นคำสั่งของเมธอดหลัก
(9) Ex_10_6 obj = new Ex_10_6(); //ประกาศออบเจกต์ตัวเอง
(10) obj.inputData(); //เรียกใช้เมธอดรับข้อมูล
(11) obj.square(number); //เรียกใช้เมธอดค่านวน โดยผ่านพารามิเตอร์
(12) obj.showResult(); //เรียกใช้เมธอดแสดงผล
(13) } //จบเมธอดหลัก
(14) public void inputData()throws IOException{ //เริ่มต้นเมธอดรับข้อมูล
(15) System.out.println("Input Integer :"); //แสดงข้อความให้ผู้ใช้เห็น
(16) number=Integer.parseInt(stdin.readLine()); //รับข้อมูลที่จำค่านวน
(17) } //จบเมธอดรับข้อมูล
(18) public void square(int n){ //เริ่มต้นเมธอดค่านวน โดยกำหนดพารามิเตอร์
(19) square = n*n; //ค่านวนตามพารามิเตอร์ที่รับมา
(20) } //จบเมธอดค่านวน
(21) public void showResult(){ //เริ่มต้นเมธอดแสดงผล
(22) System.out.println("The Square of "+number+" :"+square); //แสดงผลการค่านวน
(23) } //จบเมธอดแสดงผล
(24) } //จบคลาส

```

ผลการรัน จะให้ผู้ใช้ป้อนตัวเลขแล้ว โปรแกรมจะค่านวนค่ายกกำลังสองของเลขที่ป้อน
ดังตัวอย่าง

```
Input Integer :100
```

```
The Square of 100 :10000
```

ตัวอย่างที่ 10.7 จงเขียนโปรแกรมเก็บค่าอาร์เรย์ดังนี้ 5, 10, 15, 20, 25 หลังจากนั้นผ่านค่าอาร์เรย์เข้าไปในเมธอดเพื่อแปลงเป็นเลขยกกำลังสองแล้วแสดงผล

รหัสต้นฉบับ :

```
(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_7 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3) InputStreamReader reader = new InputStreamReader(System.in);
(4) BufferedReader stdin = new BufferedReader(reader);
(5) static int number[]={5, 10, 15, 20, 25}; //ประกาศอาร์เรย์และกำหนดค่าตามที่โจทย์ต้องการ

(6) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(7) { //เริ่มต้นคำสั่งของเมธอดหลัก
(8) Ex_10_7 obj = new Ex_10_7(); //ประกาศออบเจกต์ตัวเอง
(9) System.out.println("Data Befor Convert:"); //แสดงข้อความให้ผู้ใช้เห็น
(10) obj.showResult(); //เรียกใช้เมธอดแสดงผลข้อมูล
(11) obj.convertArray(number); //เรียกใช้เมธอดเพื่อคำนวณโดยผ่านค่าอาร์เรย์
(12) System.out.println("Data After Convert:"); //แสดงข้อความให้ผู้ใช้เห็น
(13) obj.showResult(); //เรียกใช้เมธอดแสดงผลข้อมูลหลังคำนวณ
(14) } //จบเมธอดหลัก
(15) public void convertArray(int n[]){ //เริ่มต้นเมธอดสำหรับคำนวณ
(16) for(int i=0;i<n.length;i++) //วนลูปตามขนาดของอาร์เรย์ที่ผ่านพารามิเตอร์เข้ามา
(17) n[i]*=n[i]; //คำนวณเป็นค่ายกกำลังสองทีละรายการ
(18) } //จบเมธอดคำนวณ
(19) public void showResult(){ //เริ่มเมธอดแสดงผล
(20) for(int i=0;i<5;i++) //วนลูปเท่ากับจำนวนข้อมูล
(21) System.out.println(number[i]); //แสดงข้อมูลที่ทีละรายการ
(22) } //จบเมธอดแสดงผล
(23) } //จบคลาส
```

ผลการรัน จะแสดงตัวเลขก่อนและหลังการคำนวณ แสดงตัวอย่างการรันดังนี้

Data Befor Convert:

5
10
15
20
25

Data After Convert:

25
100
225
400
625

ตัวอย่างที่ 10.8 จงเขียน โปรแกรมคำนวณหาพื้นที่ของสี่เหลี่ยมที่ผู้ใช้ป้อนความกว้างและความยาว โดยใช้เมธอดที่มีผ่านค่าเพื่อการคำนวณ

รหัสต้นฉบับ :

```
(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_8 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3) InputStreamReader reader = new InputStreamReader(System.in);
(4) BufferedReader stdin = new BufferedReader(reader);
(5) static int length, high,area; //ประกาศตัวแปรความยาว ความสูง และพื้นที่
(6) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(7) { //เริ่มต้นคำสั่งของเมธอดหลัก
(8) Ex_10_8 obj = new Ex_10_8(); //ประกาศออบเจกต์ตัวเอง
(9) obj.inputLength(); //เรียกใช้เมธอดรับความยาว
(10) obj.inputHight(); //เรียกใช้เมธอดรับความสูง
(11) obj.caculateArea(length, high); //เรียกใช้เมธอดคำนวณ โดยผ่านความยาวและความสูง
(12) obj.showData(area); //เรียกใช้เมธอดแสดงพื้นที่
(13) } //จบเมธอดหลัก
```

```

(14) public void inputLength() throws IOException{ //เริ่มเมธอดรับความยาว
(15)     System.out.println("Input Length :"); //แสดงข้อความให้ผู้ใช้เห็น
(16)     length=Integer.parseInt(stdin.readLine()); //รับความยาวเก็บใน length
(17) } //จบเมธอดรับความยาว
(18) public void inputHigh()throws IOException{ //เริ่มเมธอดรับความสูง
(19)     System.out.println("Input High :"); //แสดงข้อความให้ผู้ใช้เห็น
(20)     high=Integer.parseInt(stdin.readLine()); //รับความสูงเก็บใน high
(21) } //จบเมธอดรับความสูง
(22) public void caculateArea(int l, int h){ //เริ่มต้นเมธอดคำนวณผ่านความยาวและความสูง
(23)     area =l*h; //คำนวณพื้นที่เก็บที่ area
(24) } //จบเมธอดคำนวณ
(25) public void showData(int a){ //เริ่มเมธอดแสดงข้อมูลโดยผ่านค่าที่จะแสดง
(26)     System.out.println("The area is "+a); //แสดงผลตามที่ผ่านค่าเข้ามา
(27) } //จบเมธอดแสดงผล
(28) } //จบคลาส

```

ผลการรันจะให้ป้อนความยาวและความสูง แล้วแสดงผลการคำนวณพื้นที่ตามที่สุดรระบุไว้แสดงผลดังนี้

```

Input Length :50
Input Hight :20
The area is 1000

```

ตัวอย่างที่ 10.9 จงเขียน โปรแกรมคำนวณราคาค่าปรับอัตราการลงทะเบียนซ้ำ ต่อวัน วันละ 50 บาท โดยให้ผู้ใช้ป้อนจำนวนวันที่ลงทะเบียนซ้ำ

รหัสต้นฉบับ :

```

(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_9 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3)     InputStreamReader reader = new InputStreamReader(System.in);
(4)     BufferedReader stdin = new BufferedReader(reader);

```

```

(5) static int day,fee; //ประกาศตัวแปรจำนวนวันและค่าปรับ
(6) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(7) { //เริ่มต้นคำสั่งของเมธอดหลัก
(8)     Ex_10_9 obj = new Ex_10_9(); //ประกาศออบเจกต์ตัวเอง
(9)     obj.inputDay(); //เรียกใช้เมธอดรับจำนวนวันและค่าปรับ
(10)    obj.showData(fee); //เรียกใช้เมธอดสำหรับ
(11) } //จบเมธอดหลัก
(12) public void inputDay()throws IOException{ //เริ่มเมธอดรับข้อมูล
(13)     System.out.println("Input Day :"); //แสดงข้อความให้ผู้ใช้เห็น
(14)     day=Integer.parseInt(stdin.readLine()); //รับจำนวนวันเก็บใน day
(15)     calculateFee(day); //เรียกใช้เมธอดสำหรับคำนวณค่าปรับ
(16) } //จบเมธอดรับข้อมูล
(17) public void calculateFee(int d){ //เริ่มเมธอดคำนวณ
(18)     fee =d*50; //คำนวณค่าปรับเก็บใน fee
(19) } //จบเมธอดคำนวณ
(20) public void showData(int f){ //เริ่มเมธอดแสดงผล
(21)     System.out.println("The total of free chart :"+f); //แสดงค่าที่ผ่านเข้ามา
(22) } //จบเมธอดแสดงผล
(23) } //จบคลาส

```

ผลการรัน จะให้ผู้ใช้ป้อนจำนวนวัน แล้วโปรแกรมจะคำนวณและแสดงผล ตัวอย่างดังนี้

```
Input Day :5
```

```
The total of free chart :250
```

ตัวอย่างที่ 10.10 จงเขียน โปรแกรมเพื่อให้ผู้ใช้ป้อนราคาสินค้าจำนวน 3 รายการแล้วให้เขียนโปรแกรมเพื่อหาค่าสินค้าที่มีราคาสูงสุด แล้วรายงานผล

รหัสต้นฉบับ :

```

(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_10 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล

```

```

(3) InputStreamReader reader = new InputStreamReader(System.in);
(4) BufferedReader stdin = new BufferedReader(reader);
(5) static int good[]=new int[3];           //ประกาศตัวแปรเพื่อเก็บราคาสินค้า
(6) int max;                               //ประกาศตัวแปรเพื่อเก็บค่ามากที่สุด
(7) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(8) {                                       //เริ่มต้นคำสั่งของเมธอดหลัก
(9)     Ex_10_10 obj = new Ex_10_10();     //ประกาศออบเจกต์ตัวเอง
(10)    obj.inputData();                   //เรียกใช้เมธอดรับข้อมูล
(11)    obj.findMaximum(good[0],good[1],good[2]); //เรียกใช้เมธอดหาค่าสูงสุด
(12)    obj.showData();                   //เรียกใช้เมธอดแสดงผล
(13) }                                     //จบเมธอดหลัก
(14) public void inputData() throws IOException { //เริ่มเมธอดรับข้อมูล
(15)     for(int i=0;i<3;i++){             //วนลูปรับข้อมูล
(16)         System.out.println("Goods price "+(i+1)+" :"); //แสดงข้อความให้ผู้ใช้เห็น
(17)         good[i]=Integer.parseInt(stdin.readLine()); //รับข้อมูลสินค้าทีละรายการ
(18)     }                                 //จบลูป
(19) }                                    //จบเมธอดรับข้อมูล
(20) public void findMaximum(int x,int y, int z){ //เริ่มเมธอดหาค่าสูงสุด
(21)     max=x;                             //กำหนดค่าสูงสุดเท่ากับค่าที่ผ่านเข้าตัวแรก
(22)     if(max<y) max=y;                   //เปรียบเทียบหากค่าว่าค่าสูงสุดเป็นตัวที่สองหรือไม่
(23)     if(max<z) max=z;                   //เปรียบเทียบหากค่าว่าค่าสูงสุดเป็นตัวที่สามหรือไม่
(24) }                                     //จบเมธอดคำนวณ
(25) public void showData(){               //เริ่มเมธอดแสดงผล
(26)     System.out.println("The Maximum is "+max); //แสดงผล
(27) }                                     //จบเมธอด
(28) }                                    //จบคลาส

```

ผลการรัน จะให้ผู้ใช้ป้อนราคาสินค้า 3 ชนิด แล้ว โปรแกรมจะแสดงค่า ดังตัวอย่างด้านล่าง

Goods price 1 :500

Goods price 2 :700

Goods price 3 :300

The Maximum is 700

เมธอดแบบคืนค่า

พนิดา พาณิชกุล (2554, หน้า 223) กล่าวว่า หลังการดำเนินการตามคำสั่งของเมธอดจนได้ผลลัพธ์แล้วการส่งค่าผลลัพธ์กลับคืนจะใช้คำสั่ง “return” ซึ่งมีลักษณะการทำงานคือ

1. ใช้จบการทำงานของเมธอด ถ้าโปรแกรมพบคำสั่ง return เมื่อใดก็จบการทำงานของเมธอดนั้นทันที
2. ถ้าหลังคำสั่ง return มีตัวแปรใดตามมาก่อนจะส่งค่าตัวแปรนั้นให้กับเมธอดที่เรียกมา
3. ถ้ามีการส่งค่ากลับ ชนิดของข้อมูลจะต้องเป็นชนิดเดียวกันกับที่ระบุไว้ใน data_type ของ Header ของเมธอดนั้น

แอลเลน บี. คอว์นี่ (Allen B. Downey, 2011, p. 32) กล่าวว่า เมธอดบางเมธอดจะให้ค่ากลับคืนเช่นเมธอดที่เรียก Math แต่ในกรณีที่บางเมธอด ก็จะดำเนินการโดยไม่ส่งค่ากลับคืน เช่น println เป็นต้น

ดังนั้น เมธอดแบบมีการคืนค่าเป็นลักษณะของงานที่เป็นเมธอดที่ใช้กันเป็นส่วนมาก เพราะลักษณะของเมธอดนั้นจะต้องเรียกใช้งานแล้วได้ผลลัพธ์ออกมาค่าหนึ่ง หมายความว่า การสั่งให้ เมธอดทำงานแล้วจะต้องได้ผลการประมวลผลออกมา เพื่อใช้ประโยชน์อย่างใดอย่างหนึ่งจากค่าที่ได้จากการทำงานของเมธอด ลักษณะการใช้งานเมธอดจะต้องหาตัวแปรชนิดเดียวกันกับค่าที่คืนออกมานั้นมารับค่าที่คืนออกมาจากเมธอด สิ่งที่สำคัญและแตกต่างจากเมธอดที่ไม่มีการคืนค่าคือ จะมีชนิดของข้อมูลที่คืนกลับมา ตอนท้ายของเมธอดจะต้องมีคำสั่ง return ตามด้วยค่าที่ต้องการคืนให้ตรงกับชนิดที่ระบุไว้ การพิจารณาว่าเมธอดใดคืนค่าใดให้ดูที่ส่วนหัวเมธอดที่จะระบุชนิดของข้อมูลที่จะคืนออกมา

สิ่งที่สำคัญและแตกต่างจากเมธอดที่ไม่มีการคืนค่า คือ จะมีชนิดของข้อมูลที่คืนกลับมา และคำสั่ง return ตามด้วยค่าที่ต้องการคืนให้ตรงกับชนิดที่ระบุไว้ในส่วนแรกของเมธอด

การพิจารณาว่าเมธอดใดคืนค่าใดให้ดูที่ส่วนหัวเมธอดที่จะระบุชนิดของข้อมูลที่จะคืนออกมา เช่น

```
public int mytest(float x)
{
    System.out.println("I've worked");
    return (int)x;
```

```
}

```

public int mytest(float x); เมธอด mytest จะคืนค่าจำนวนเต็มออกมาซึ่งการใช้งานจะต้องหาตัวแปรที่เป็นจำนวนเต็มมารับค่า เช่น int k=mytest(20.5); โดยที่ตัวแปร k จะรับค่าที่คืนออกมาเป็นต้น

return (int)x; เป็นการคืนค่าเป็น integer ที่แปลงค่า float x ให้เป็นจำนวนเต็มตามที่ต้องการ การเรียกใช้จะใช้

int m=mytest(22.77); เรียกใช้โดยผ่านค่า 22.77 และแปลงให้เป็น จำนวนเต็มแล้วเก็บผลลัพธ์ไว้ใน m

การเรียกใช้เมธอดแบบคืนค่าอาจเรียกใช้ได้โดยตรงโดยไม่ต้องมีตัวแปรมารับค่าที่คืนหากเป็นการแสดงผล เช่น ถ้าต้องการเรียกใช้ mytest ดังกล่าวให้แสดงผล ได้ดังนี้

System.out.format(“%d”,mytest(22.77)); ซึ่งสามารถแสดงจำนวนเต็มที่ผ่านการแปลงแล้วได้ทันทีโดยไม่ต้องมีตัวแปรมารับค่าอีกก็ได้

ตัวอย่างที่ 10.11 จงเขียน โปรแกรมรับค่าตัวเลขจำนวนหนึ่งแล้วแสดงผลเลขยกกำลังสองของเลขจำนวนนั้นที่ป้อนเข้าไป

รหัสต้นฉบับ :

```
(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_11 { //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3) InputStreamReader reader = new InputStreamReader(System.in);
(4) BufferedReader stdin = new BufferedReader(reader);
(5) static int number,square; //ประกาศตัวแปรเพื่อรับข้อมูลและค่ายกกำลังสอง
(6) public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(7) { //เริ่มคำสั่งของเมธอดหลัก
(8) Ex_10_11 obj = new Ex_10_11(); //ประกาศออบเจกต์ของตนเอง
(9) obj.inputNumber(); //เรียกใช้เมธอดรับข้อมูล
(10) square=calculateSquare(number); //เรียกใช้เมธอดคำนวณและคืนค่าผลการคำนวณ
(11) obj.showData(); //เรียกใช้เมธอดแสดงผล
(12) } //จบเมธอดหลัก
(13) public void inputNumber()throws IOException{ //เริ่มเมธอดรับข้อมูล
```

```

(14) System.out.println("Input Integer :");           //แสดงให้ผู้ใช้เห็น
(15) number=Integer.parseInt(stdin.readLine());      //รับค่าเก็บใน number
(16) }                                               //จบเมธอดรับข้อมูล
(17) public static int calculateSquare(int n){       //เริ่มเมธอดคำนวณ
(18)     n*=n;                                       //คำนวณและเก็บค่าใน n
(19)     return n;                                   //คืนค่าผลการคำนวณ
(20) }                                               //จบเมธอดคำนวณ
(21) public void showData(){                         //เริ่มเมธอดแสดงผล
(22)     System.out.println("The square is "+square); //แสดงผล
(23) }                                               //จบเมธอดแสดงผล
(24) }                                               //จบคลาส

```

ผลการรันโปรแกรมนี้ จะให้ผู้ใช้ป้อนตัวเลข แล้วโปรแกรมจะคำนวณค่ายกกำลังสองออกมาให้ แสดงตัวอย่างผลการรันดังนี้

```

Input Integer :50
The square is 2500

```

ตัวอย่างที่ 10.12 จงเขียน โปรแกรมแปลงค่ากิโลกรัมเป็นปอนด์โดย 1 ก.ก. เท่ากับ 2.205 ปอนด์
รหัสต้นฉบับ :

```

(1) import java.io.*;                               //ส่วนการนำแพ็คเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_12 {                          //เริ่มต้นคลาส
    //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(3)     InputStreamReader reader = new InputStreamReader(System.in);
(4)     BufferedReader stdin = new BufferedReader(reader);
(5)     static int kg;                               //ประกาศตัวแปรตัวเลขเป็นกิโลกรัม
(6)     static double pound;                         //ประกาศตัวแปรปอนด์
(7)     public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(8)     {                                             //เริ่มต้นคำสั่งของเมธอดหลัก
(9)         Ex_10_12 obj = new Ex_10_12();         //ประกาศออบเจกต์ตัวเอง
(10)        obj.inputNumber();                       //เรียกใช้เมธอดรับข้อมูล

```



```

(8)   Ex_10_13 obj = new Ex_10_13();      //ประกาศออบเจกต์ตัวเอง
(9)   obj.inputString();                  //เรียกใช้เมธอดรับข้อมูล
(10)  sum=calSum(number1,number2);        //เรียกใช้เมธอดคำนวณและคืนค่าเก็บใน sum
(11)  System.out.println("Sumation is "+sum); //แสดงผล
(12)  }                                    //จบเมธอดหลัก
(13)  public void inputString()throws IOException{ //เริ่มเมธอดแสดงผล
(14)  System.out.println("Input Begining Number :"); //แสดงข้อความให้ผู้ใช้เห็น
(15)  number1=Integer.parseInt(stdin.readLine()); //รับค่าตัวเลขเก็บใน number1
(16)  System.out.println("Input Ending Number :"); //แสดงข้อความให้ผู้ใช้เห็น
(17)  number2=Integer.parseInt(stdin.readLine()); //รับค่าตัวเลขเก็บใน number2
(18)  }                                    //จบเมธอดรับข้อมูล
(19)  public static int calSum(int begin, int end){ //เริ่มเมธอดคำนวณ
(20)  int s=0;                               //ประกาศตัวแปรเพื่อคำนวณ
(21)  for(int i=begin;i<=end;i++)           //วนลูปคำนวณตามที่โจทย์ต้องการ
(22)  s+=i;                                  //คำนวณ
(23)  return s;                             //คืนค่า s ไปยังเมธอดที่เรียกใช้
(24)  }                                    //จบเมธอดคำนวณ
(25)  }                                    //จบคลาส

```

ผลการรัน จะให้ผู้ใช้ป้อนตัวเลขเริ่มต้นและสิ้นสุด หลังจากนั้น โปรแกรมจะประมวลผล ตัวอย่างผลการรันได้ดังนี้

```

Input Begining Number :1
Input Ending Number :10
Summation is 55

```

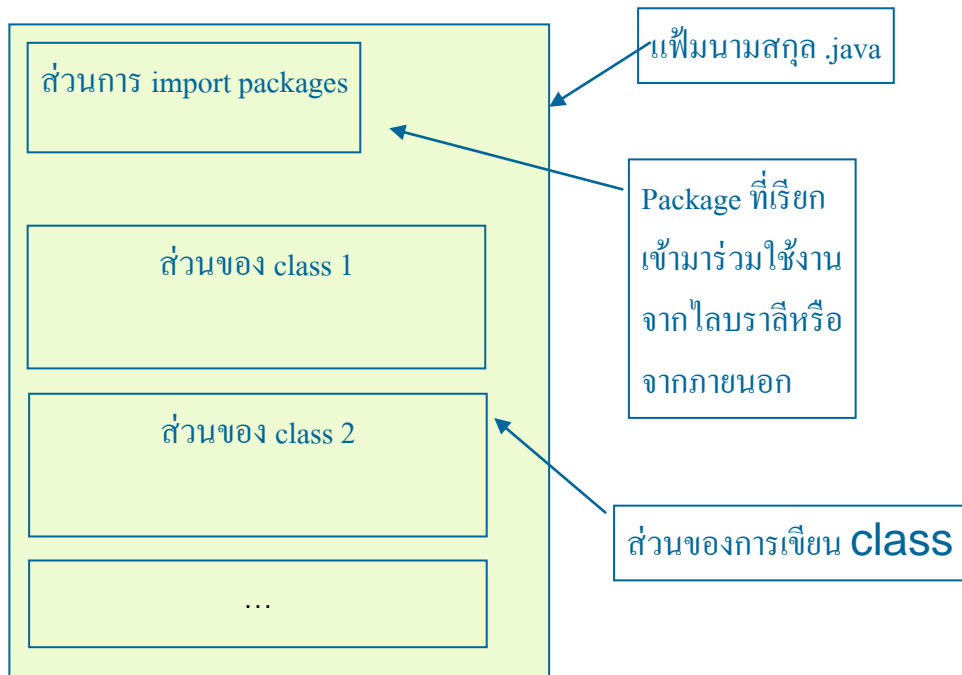
การใช้งานคลาส

บริษัทซันไมโครซิสเต็ม (Sun Microsystems ,2007) กล่าวว่า คลาสคือพิมพ์เขียวซึ่งจะต้องถูกสร้างขึ้นมาจากคลาสเดี่ยว วิกิพีเดีย (Wikipedia, 2007) กล่าวว่า คลาส คือ การเขียนโปรแกรมภาษาคอมพิวเตอร์ซึ่งจัดกลุ่มเกี่ยวกับฟิลด์และเมธอด พนิดา พานิชกุล (2548, หน้า 7) กล่าวว่าคลาสคือ กลุ่มของวัตถุที่ลักษณะคล้ายกันตามที่ให้ความคิดรวบยอดไว้ (Concept)

ยาคอบ เฟน (Yacov Fain, 2007, p. 215-216) กล่าวว่า คลาส คือ แม่แบบ (Template) หรือพิมพ์เขียว (Blueprint) ซึ่งกำหนดว่าวัตถุ (Object) จะมีข้อมูล (Data) และเมธอดอะไร

คลาส คือ โปรแกรมที่สร้างขึ้นมาเพื่อใช้งานโดยแนวคิดดังกล่าวนี้จะกล่าวอีกครั้งในบทที่ 13 โดยจะกล่าวถึงการเขียนโปรแกรมโดยใช้คลาสเดี่ยวหรือหลายคลาส แต่ในบทนี้ยังไม่เน้นการเขียนโปรแกรมแบบเชิงวัตถุ การเขียนโปรแกรมเพื่อเรียกใช้คลาสนั้นยังคงเป็นการเรียกใช้ในแนวของโปรแกรมแบบโครงสร้าง เท่านั้น โครงสร้างของรหัสต้นฉบับของภาษาจาวาแสดงดังภาพที่

10.2



ภาพที่ 10.2 แสดงโครงสร้างรหัสต้นฉบับของภาษาจาวา

ตัวอย่างโครงโปรแกรม เช่น

```
import java.io.*;

public class firstClass
{
    public static void main(String args[])
    {
        // เนื้อหาของโปรแกรม
    }
}
```

โครงสร้างของคลาสแบบมากกว่าหนึ่งคลาส

```
import java.io.*;
public class firstClass
{
    public static void main(String args[])
    {
        // เนื้อหาของ โปรแกรมหลัก
    }
}
// class ที่สอง
class secondClass
{
    public void metA(void)
    {
        // เนื้อหาของ โปรแกรม method A
    }
}
```

ตัวอย่างโครงสร้างของคลาสเดี่ยว

```
import java.io.*;
public class firstClass
{
    public static void main(String args[])
    {
        System.out.println("Hello world");
    }
}
```

การสร้างคลาสนั้นสามารถสร้างไว้ที่ภายในแฟ้มเดียวกันได้ ซึ่งคลาสหลักที่จะใช้รันโปรแกรมจะต้องมีเมธอด main อยู่ด้วยเสมอ ส่วนคลาสอื่น ๆ จะต้องสร้างคลาสและเรียกใช้โดย

การสร้างอ็อบเจกต์ของคลาสนั้นขึ้นมาก่อนแล้วจึงเรียกใช้เมธอดภายในตามที่ต้องการ ยาคอบ (Yacov Fain, 2007, p. 219) กล่าวว่า โดยปกติแล้วเจวีเอ็ม (JVM: Java Virtual Machine) จะเรียกเมธอด main() ของคลาสหลักสำหรับใช้งาน ซึ่งหากมีใช้คลาสหลักเพียงคลาสเดียวและต้องมี main() ตรงตามข้อกำหนดในเรื่องชื่อไฟล์ตรงกับชื่อคลาสอีกด้วย ตัวอย่างต่อไปนี้แสดงการสร้างคลาสและเรียกใช้งานเพียงโปรแกรมเดียว

ตัวอย่างที่ 10.14 จงเขียนโปรแกรมรับค่าตัวเลข 2 ค่า โดยกำหนดค่าเริ่มต้นและค่าสุดท้าย แล้วหาผลรวมระหว่างตัวเลขและตัวสุดท้ายแล้วแสดงผล

รหัสต้นฉบับ :

```
(1) import java.io.*; //ส่วนการนำแพคเกจ java.io เข้ามาร่วมใช้งาน
(2) public class Ex_10_14 { //เริ่มต้นคลาสแรก
(3)     public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(4)     { //เริ่มต้นคำสั่งของเมธอดหลัก
(5)         calNumber obj = new calNumber(); //ประกาศตัวแปรออบเจกต์ของคลาสที่สอง
(6)         obj.inputString(); //เรียกใช้เมธอดรับข้อมูล
(7)         System.out.println("Summation is "+obj.getSum()); //เรียกใช้เมธอดเพื่อแสดงผล
(8)     } //จบเมธอดหลัก
(9) } //จบคลาสแรก
(10) class calNumber{ //เริ่มต้นคลาสที่สอง
        //ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล
(11)     InputStreamReader reader = new InputStreamReader(System.in);
(12)     BufferedReader stdin = new BufferedReader(reader);
(13)     int number1,number2,sum; //ประกาศตัวแปรที่ต้องการใช้
(14)     public void inputString()throws IOException{ //เริ่มเมธอดรับข้อมูล
(15)         System.out.println("Input Beginning Number :"); //แสดงข้อความให้ผู้ใช้เห็น
(16)         number1=Integer.parseInt(stdin.readLine()); //รับข้อมูลเก็บใน number1
(17)         System.out.println("Input Ending Number :"); //แสดงข้อความให้ผู้ใช้เห็น
(18)         number2=Integer.parseInt(stdin.readLine()); //รับข้อมูลเก็บใน number2
(19)     } //จบเมธอดรับข้อมูล
(20)     public int calSum(int begin, int end){ //เริ่มเมธอดคำนวณ
```


(21)	for(int i=begin;i<=end;i++)	//วนลูปคำนวณตามที่โจทย์ต้องการ
(22)	sum+=i;	//คำนวณ
(23)	return sum;	//คืนค่าผลรวมไปยังผู้ที่เรียกใช้
(24)	}	//จบเมธอดแสดงผล
(25)	public int getSum(){	//เริ่มเมธอดคืนค่าไปแสดงผล
(26)	return calSum(number1,number2);	//เรียกใช้เมธอดคำนวณและคืนค่าผลที่ได้
(27)	}	//จบเมธอดคืนค่าผลผลรวม
(28)	}	//จบคลาส

ผลการรัน จะคล้ายกับตัวอย่างที่ 10.13 ดังนี้

```
Input Beginning Number :1
Input Ending Number :10
Summation is 55
```

ตัวอย่างที่ 10.15 จงเขียนโปรแกรมบวกเลขโดยใช้เมธอดหาผลบวกกับเลขที่ป้อนเข้ากับ 100 รหัสต้นฉบับ :

(1)	import java.io.*;	//ส่วนการนำแพ็คเกจ java.io เข้ามาร่วมใช้งาน
(2)	public class Ex_10_15 {	//เริ่มต้นคลาสแรก
(3)	public static void main(String args[]) throws IOException	//ส่วนของเมธอดหลัก
(4)	{	//เริ่มต้นคำสั่งของเมธอดหลัก
(5)	plusNumber obj = new plusNumber();	//ประกาศออบเจกต์ของคลาสที่สอง
(6)	obj.inputString();	//เรียกใช้เมธอดรับข้อมูล
(7)	System.out.println("The result is "+obj.getSum());	//เรียกใช้เมธอดแสดงผล
(8)	}	//จบเมธอดหลัก
(9)	}	//จบคลาสแรก
(10)	class plusNumber{	//เริ่มต้นคลาสที่สอง
	//ส่วนการประกาศตัวแปรออบเจกต์เพื่อใช้รับข้อมูล	
(11)	InputStreamReader reader = new InputStreamReader(System.in);	
(12)	BufferedReader stdin = new BufferedReader(reader);	
(13)	int number1,sum;	//ประกาศตัวแปรรับข้อมูลและเก็บผลลัพธ์


```

(12) } //จบคลาสแรก
(13) class squareNumber{ //เริ่มคลาสที่สอง
(14) public int[] calArray(int ar[]){ //เริ่มเมธอดคำนวณด้วยอาร์เรย์
(15)     for(int i=0;i<ar.length;i++) //วนลูปตามขนาดของอาร์เรย์ที่ผ่านเข้ามา
(16)         ar[i]*=ar[i]; //คำนวณทีละรายการ
(17)     return ar; //คืนค่าอาร์เรย์ทั้งสาย
(18) } //จบเมธอดคำนวณ
(19) } //จบคลาสที่สอง

```

ผลการรันโปรแกรมนี้ คือ เมื่อป้อนค่าเข้าไปแล้ว โปรแกรมจะทำการยกกำลังสองค่าที่ป้อนเข้าไป แล้วทำการแสดงผล ในที่นี้การแสดงผลนั้นจะพบว่าค่า number นั้นได้ถูกเปลี่ยนแปลงไปเหมือนกับการเอา number มายกกำลังสองโดยตรงแต่เป็นการผ่านค่า ตำแหน่งที่อยู่ของ number เข้าไปในเมธอด square เท่านั้น ซึ่ง x ที่มารับค่า number นั้นก็จะนำเอาค่าของ ตำแหน่งที่เก็บค่า number ไปดำเนินการจึงทำให้ค่าของ number เปลี่ยนแปลงไป ตามการดำเนินการของเมธอด

การสร้างคลาสและเรียกใช้โดยไม่ต้องสร้างออบเจกต์ โดยการกำหนดเมธอดเป็น static เมธอด ดังนี้

```
public static void ...
```

```
public static int ...
```

เป็นต้น ตัวอย่างที่ 10.17 แสดงตัวอย่างดังกล่าว

ตัวอย่างที่ 10.17 จงเขียนโปรแกรมยกกำลังสองของตัวเลขที่ผู้ใช้ป้อนเข้าไป แล้วแสดงผลโดยใช้การผ่านค่าให้กับเมธอดโดยใช้การอ้างอิง โดยใช้ static เมธอด

รหัสต้นฉบับ :

```

(1) public class Ex_10_17 { //เริ่มต้นคลาส
(2)     public static void main(String args[]) throws IOException //ส่วนของเมธอดหลัก
(3)     { //เริ่มต้นคำสั่งของเมธอดหลัก
(4)         int number[]={1,2,3,4,5}; //ประกาศตัวแปรอาร์เรย์และกำหนดค่าเริ่มต้น
(5)         int result[]=new int[5]; //ประกาศตัวแปรอาร์เรย์เก็บผลลัพธ์
(6)         result=squareNumber.calArray(number); //เริ่มเมธอดคำนวณด้วยอาร์เรย์
(7)         for(int i=0;i<5;i++) //วนลูปเพื่อแสดงผล

```

```

(8)      System.out.println("Array Num :"+(i)+" : "+result[i]); //แสดงผล
(9)      } //จบเมธอดหลัก
(10)     } //จบคลาสแรก
(11)     class squareNumber{ //เริ่มคลาสที่สอง
(12)     public static int[] calArray(int ar[]){ //เริ่มเมธอดคำนวณด้วยอาร์เรย์
(13)     for(int i=0;i<ar.length;i++) //วนลูปตามขนาดของอาร์เรย์ที่ผ่านเข้ามา
(14)     ar[i]*=ar[i]; //คำนวณทีละรายการ
(15)     return ar; //คืนค่าอาร์เรย์ทั้งสาย
(16)     } //จบเมธอดคำนวณ
(17)     } //จบคลาสที่สอง

```

ผลการรันจะได้เช่นเดียวกันกับที่ผ่านมา แต่สิ่งที่เป็นข้อสังเกตคือการเรียกใช้งานโดยไม่ต้องสร้างวัตถุ (new) ขึ้นมาใหม่ เหมาะสมกับการใช้งานบ่อย ๆ เช่นคลาส Math, String, เป็นต้น

สรุป

เมธอด คือ การแบ่งการทำงานออกเป็นส่วนย่อย ๆ หรือเป็น โมดูล เมธอดแบ่งออกเป็นหลายประเภทขึ้นอยู่กับเกณฑ์ของการแบ่ง เช่น แบ่งตามตำแหน่ง แบ่งตามการคืนค่า หรือแบ่งตามการผ่านค่าพารามิเตอร์ โดยทั่วไปประกอบไปด้วยส่วนหัว ตัวหรือกลุ่มคำสั่งและส่วนการคืนค่าคือ return เมธอดแบบไม่คืนค่าจะนำหน้าเมธอดด้วย void และเรียกใช้งานโดยเรียกใช้โดยระบุชื่อเมธอดโดยไม่ต้องมีตัวแปรมารับค่าที่คืนกลับมา เมธอดแบบคืนค่า คือ เมธอดที่นำหน้าด้วยชนิดของข้อมูล ที่ไม่ใช่ void โดยเมื่อเรียกใช้จะต้องมีตัวแปรประเภทเดียวกันกับค่าที่คืนกลับมา ยกเว้นคำสั่งแสดงผล และการสร้างจะต้องใช้คำสั่ง return ตามด้วยค่าที่เป็นชนิดเดียวกันกับที่ระบุไว้ที่หัวเมธอด การเรียกใช้เมธอดอาจเรียกใช้โดยการผ่านค่าพารามิเตอร์ นอกจากนั้นการแบ่งโปรแกรมออกเป็นคลาสย่อย ๆ เป็นอีกวิธีการที่ทำให้โปรแกรมที่มีความซับซ้อนสูงลดความซับซ้อนลง โดยอาจแบ่งไว้ในไฟล์เดียวกันหรือต่างไฟล์ก็ได้

แบบฝึกหัดท้ายบท

ตอนที่ 1 จงเขียนเครื่องหมาย ✓ หน้าข้อที่กล่าวถูกต้อง และ เขียนเครื่องหมาย ✗ หน้าข้อที่กล่าวผิด

1. _____ เมธอดเปรียบเสมือนเป็นฟังก์ชันเพื่อแบ่งย่อยโปรแกรม
2. _____ การสร้างเมธอดจำเป็นเสมอในการเขียน โปรแกรมแบบโครงสร้าง
3. _____ การสร้างเมธอดจำเป็นเสมอในการเขียน โปรแกรมเชิงวัตถุ
4. _____ เมธอดที่ไม่มีการคืนค่าต้องนำหน้าด้วย void
5. _____ public int my() เมธอดนี้ return '0' จะ Error
6. _____ public static void main() เป็นฟังก์ชันที่อยู่ตลอดการรัน โปรแกรม
7. _____ public void my(String s) เมธอดนี้จะเรียกใช้โดยผ่าน String
8. _____ คลาสโปรแกรมทุก ๆ โปรแกรมที่เขียนด้วยจาวา
9. _____ คลาสประกอบด้วยเมธอดเสมอ
10. _____ แพคเกจของภาษาจาวาหนึ่งแพคเกจ สามารถมีคลาสได้หลายคลาส

ตอนที่ 2 จงตอบคำถามต่อไปนี้

1. จงเขียนโปรแกรมขีดเส้นบนจอภาพ โดยกำหนดให้เป็นเมธอดแล้วป้อนค่าอักขระที่จะเขียนบนจอภาพได้ด้วย เช่นคำสั่ง `line("=");` จะขีดเส้นต่อกัน 60 คอลัมน์
2. จงเขียนโปรแกรมที่สามารถหาค่าผลคูณของเลขเป็นเลขยกกำลังสองของตัวเลขที่ป้อน
3. จงเขียนโปรแกรมเปลี่ยนค่าองศาเซลเซียส เป็นองศาฟาเรนไฮต์ หรือ จากองศาฟาเรนไฮต์ เป็นองศาเซลเซียส โดยให้ป้อนค่าแล้วเลือกว่าจะเปลี่ยนจาก องศาฟาเรนไฮต์เป็นองศาเซลเซียสหรือจากองศาเซลเซียสเป็นองศาฟาเรนไฮต์
4. จงเขียน โปรแกรมป้อนค่าตัวเลข 3 ค่าแล้ว ใช้เมธอดหาค่าที่น้อยที่สุดโดยการคืนค่า
5. จงเขียนโปรแกรมรับชื่อและคะแนนของนักศึกษาจำนวน 5 คน แล้วคำนวณหาคะแนนเฉลี่ยโดยใช้เมธอด
6. จงเปรียบเทียบข้อดีข้อเสียของการเขียน โปรแกรมแบบใช้เมธอดและไม่ใช้เมธอด
7. จงเปรียบเทียบข้อดีข้อเสียของการเขียน โปรแกรมแบบใช้เมธอดที่ผ่านค่าพารามิเตอร์และไม่ใช้การผ่านค่าพารามิเตอร์ให้กับเมธอด
8. จงบอกข้อเสียของการผ่านค่าพารามิเตอร์โดยการระบุค่าตัวเลขโดยตรง
9. หากท่านต้องเขียน โปรแกรมควบคุมหัวจ่ายน้ำมันจากปั้มน้ำมันจะออกแบบให้มีเมธอดอะไรบ้าง เพราะเหตุใด
10. จงอธิบายว่าท่านมีหลักเกณฑ์ในการแบ่งเมธอดอย่างไร เพราะเหตุใด

เอกสารอ้างอิง

- จันทร์จิรา สีนทนะโยธิน และวิศรุต พลสิทธิ. (2007). “การเขียนโปรแกรมภาษา C เบื้องต้น (Introduction to C Programming)”. [Online]. Available: <http://www.vcharkarn.com/include/article/showarticle.php?Aid=18065&page=4>.
- ธีรวัฒน์ ประกอบผล. (2545). การโปรแกรมภาษาซีสำหรับงานวิทยาศาสตร์. กรุงเทพฯ: ศ.ศ.ท.
- _____. (2552). คู่มือการเขียนโปรแกรมภาษา Java. กรุงเทพฯ: ชัคเซส มีเดีย.
- _____. (2553). คู่มือการเขียนโปรแกรมภาษา Java. กรุงเทพฯ: ชิมพลิฟลาย.
- นุกูล กระจาย. (2540). การเขียนโปรแกรมในดอสและวินโดวส์ด้วยบอร์แลนด์ C++ 5.0. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- พนิดา พาณิชกุล. (2548). Object-Oriented ฉบับพื้นฐาน. กรุงเทพฯ: เคทีพี แอนด์ คอนซัลท์.
- _____. (2554). การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษาจาวา. พิมพ์ครั้งที่ 5. กรุงเทพฯ: เคทีพี แอนด์ คอนซัลท์.
- พิเชษฐ ศิริรัตนไพศาลกุล. (2553). หลักการเขียนโปรแกรมเบื้องต้นด้วยภาษา JAVA. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- Allen B. Downey.(2011). **How to Think Like a Computer Scientist Java Version**. Ventus Publishing Aps. [Online]. Available: <http://www.BookBoon.com>.
- Barry Burd. (2004). **Java FOR DUMmIES**. 2nd Edition. USA: Wiley Publishing, Inc., Indianapolis, Indiana.
- _____. (2005). **Beginning Programming with Java FOR DUMmIES**. 2ND EDITION , USA :Wiley Publishing, Inc.
- _____. (2007). **Java FOR DUMmIES**. 4th Edition. USA: Wiley Publishing, Inc., Indianapolis, Indiana.
- _____. (2011). **Java FOR DUMmIES**. 5th Edition. USA: Wiley Publishing, Inc.
- Cay Horstmann. (2002). **Object-Oriented Design & Patterns**. USA: John Wiley & Sons.
- David Etheridge. (2009 A). **Java:The Fundamentals of Objects and Classes-An Introduction to Java Programming**. Ventus Publishing Aps. [Online]. Available: <http://www.BookBoon.com>.
- Harvey M. Deitel. (2002). **Java How to Program**. 4th Edition Reviewers. USA: Prentice Hall.

- Simol Kendal. (2009). **Object Oriented Programming using Java**. Ventus Publishing Aps. [Online]. Available: <http://www.BookBoon.com>.
- Sun Microsystems.(2007). “**What Is a Class**”. [Online]. Available: <http://java.sun.com/docs/books/tutorial/java/concepts/class.html>.
- wikipedia. (2007). “**Class (computer science)**”. [Online]. Available: http://en.wikipedia.org/wiki/Class_%28computer_science%29.
- Yakov Fain. (2004). **Java™ Programming for kid, parents and grandparents**. [Online]. Available: <http://www.BookBoon.com>.