

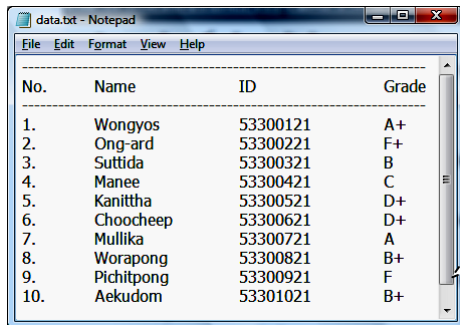
CHAPTER 06

การดำเนินการกับแฟ้มข้อมูล (File Operations)

1. ลักษณะทั่วไปของแฟ้มข้อมูล (Characteristic of File)

1. นิยามและข้อมูลเบื้องต้นของแฟ้มข้อมูล

- 1) แฟ้มข้อมูล (File / Text File) คือ แฟ้มเอกสารหรือไฟล์ที่มีรายการข้อมูลอยู่ภายใน ซึ่งจะถูกจัดเก็บไว้เป็นแฟ้มนามสกุลต่างๆ เช่น *.txt, *.dat หรือ นามสกุลอื่นๆ ซึ่งแสดงไว้ในตัวอย่างของแฟ้มข้อมูลต่อไปนี้



No.	Name	ID	Grade
1.	Wongyos	53300121	A+
2.	Ong-ard	53300221	F+
3.	Suttida	53300321	B
4.	Manee	53300421	C
5.	Kanittha	53300521	D+
6.	Choocheep	53300621	D+
7.	Mullika	53300721	A
8.	Worapong	53300821	B+
9.	Pichitpong	53300921	F
10.	Aekudom	53301021	B+

แฟ้มข้อมูลชื่อ "data.txt" สร้างขึ้นโดยโปรแกรม Notepad โดยแฟ้มนี้มีนามสกุลเป็น *.txt ซึ่งภายในเก็บผลการเรียนของนิสิตกลุ่มหนึ่ง

- 2) ข้อมูลที่ถูกเก็บในแฟ้มข้อมูลสามารถมีได้ไม่จำกัดจำนวนบรรทัด (ยาวกี่บรรทัดก็ได้)
- 3) ประเภทข้อมูลที่ถูกเก็บอยู่ในแฟ้มข้อมูลประกอบไปด้วย ตัวเลข (จำนวนเต็ม และจำนวนจริง) และสตริง
- 4) ขนาดของแฟ้มข้อมูลคำนวณได้จากจำนวนของอักขระทุกตัวที่ถูกเก็บอยู่ในแฟ้มข้อมูลนั้น (1 อักขระ = 1 Byte)

2. การดำเนินการกับแฟ้มข้อมูล

การดำเนินการกับแฟ้มข้อมูล คือ การกระทำหรือประมวลผลบางอย่างกับแฟ้มข้อมูล ซึ่งในเอกสารบทนี้จะนำเสนอ [1] การอ่านแฟ้มข้อมูล และ [2] การเขียนแฟ้มข้อมูล

- 1) การอ่านแฟ้มข้อมูล (Reading File) คือ การรับค่าหรือรับข้อมูลจากแฟ้มข้อมูลเข้ามายังเครื่องคอมพิวเตอร์ เพื่อกระทำการบางอย่าง เช่น คำนวณผล แสดงผล เป็นต้น
- 2) การเขียนแฟ้มข้อมูล (Writing File) คือ การบันทึกค่าหรือบันทึกข้อมูลจากการประมวลผลบนเครื่องคอมพิวเตอร์ไปเก็บไว้ยังแฟ้มข้อมูล

2. การอ่านและเขียนแฟ้มข้อมูล (Reading and Writing Files)

1. ขั้นตอนการอ่านแฟ้มข้อมูล

การอ่านแฟ้มข้อมูล ในภาษาจาวาสามารถทำได้หลายวิธี แต่ในบทนี้จะนำเสนอการอ่านแฟ้มข้อมูลด้วยคลาส Scanner ซึ่งมีลักษณะการทำงานเช่นเดียวกับการอ่านข้อมูลจากแป้นพิมพ์ที่กล่าวไปแล้วในบทที่ 2 โดยมีขั้นตอนต่างๆ ดังต่อไปนี้

- 1) การนำเข้าคลาส Scanner และแพ็คเกจ java.io ก่อนการเปิดอ่านแฟ้มข้อมูลจะต้องนำเข้าคลาส Scanner และแพ็คเกจ java.io ก่อนเสมอ โดยใช้คำสั่ง `import java.util.Scanner;` และคำสั่ง `import java.io.*;` ซึ่งจะนำไปก่อนเขียนหัวคลาส ดังตัวอย่าง

```

1 import java.util.Scanner;
2 import java.io.*;
3 public class Test {
4     ...
5 }

```

การ import สองคำสั่งนี้ จะให้คำสั่งใด
ขึ้นก่อนก็ได้ แต่ต้องมีให้ครบทั้ง 2 คำสั่ง

- 2) การตรวจสอบความผิดพลาดในการเปิดอ่านแฟ้มข้อมูล เมท็อดใดก็ตามที่เป็นเมท็อดสำหรับเปิดอ่านแฟ้มข้อมูล (ในที่นี้คือเมท็อด main) จะต้องมีการระบุส่วนของคำสั่ง **throws IOException** ไว้ตอนท้ายของหัวเมท็อดนั้นเสมอ เพื่อให้ระบบตรวจสอบข้อผิดพลาด เช่น ตรวจสอบว่าแฟ้มข้อมูลที่เปิดอ่านมีอยู่จริงในระบบหรือไม่ เป็นต้น ดังตัวอย่าง

```

1 import java.util.Scanner;
2 import java.io.*;
3 public class Test {
4     public static void main(String[] args) throws IOException {
5         ...
6     }
7 }

```

คำว่า throws ต้องเติม s ด้วย

- 3) การสร้างตัวอ่านจากแฟ้มข้อมูล เป็นการกำหนดชื่อตัวอ่านเพื่อใช้สำหรับอ่านค่าข้อมูลต่างๆ จากแฟ้มข้อมูล โดยจะสร้างเพียงครั้งเดียวเท่านั้น และจะใช้งานตัวอ่านนั้นตลอดทั้งโปรแกรม ซึ่งมีคำสั่งดังนี้

```
Scanner <ชื่อตัวอ่าน> = new Scanner(new File("<ชื่อแฟ้มข้อมูล>"));
```

หรือ

```
File <ชื่อตัวเรียกแฟ้มข้อมูล> = new File("<ชื่อแฟ้มข้อมูล>")
```

```
Scanner <ชื่อตัวอ่าน> = new Scanner(<ชื่อตัวเรียกแฟ้มข้อมูล>);
```

```

1 import java.util.Scanner;
2 import java.io.*;
3 public class Test {
4     public static void main(String[] args) throws IOException {
5         Scanner in = new Scanner(new File("data.txt"));
6         ...
7     }
8 }

```

เปิดอ่านแฟ้มข้อมูลชื่อ data.txt

ตัวอ่านแฟ้มข้อมูลชื่อ in

เข้ามาแทนที่คำสั่ง System.in เดิม ซึ่งจะหมายถึงแฟ้มข้อมูล

- 4) การตรวจสอบว่ายังมีข้อมูลเหลือให้อ่านในแฟ้มข้อมูลอีกหรือไม่ เมื่อประกาศและสร้างทุกอย่างตามขั้นตอนที่ 1-3 แล้ว ขั้นตอนแรกที่จะเริ่มอ่านข้อมูลตั้งแต่ต้นแฟ้มจนถึงท้ายแฟ้มนั้น จะต้องตรวจสอบก่อนเสมอว่ายังมีข้อมูลในแฟ้มเหลือให้อ่านอีกหรือไม่ ซึ่งใช้เมท็อด **hasNext()** ในการตรวจสอบ โดยคำตอบที่ได้จะเป็นค่าความจริง (Boolean) ซึ่งมีรูปแบบคำสั่งดังนี้

```

while(<ชื่อตัวอ่าน>.hasNext()) {
    ...
}

```

คำสั่งอ่านค่าข้อมูล (จะกล่าวในข้อถัดไป)

ตรวจสอบว่ายังมีข้อมูลเหลือในแฟ้มอีกหรือไม่ ถ้า
ยังเหลือ (ยัง Next ได้) ก็จงวนอ่านข้อมูลไปเรื่อยๆ
จนกว่าไม่มีข้อมูลเหลือให้อ่านได้อีก ก็จบการวนซ้ำ

- 5) การอ่านค่าข้อมูลด้วยตัวอ่านเพิ่มข้อมูล ตัวอ่านจากเพิ่มข้อมูลที่เราสร้างขึ้นสามารถอ่านค่าข้อมูลประเภทจำนวนเต็ม จำนวนจริง และสตริง โดยใช้คำสั่งและเมทอดต่อไปนี้

(1) คำสั่งอ่านค่าจำนวนเต็มประเภท int โดยใช้เมทอด nextInt()

```
int <ชื่อตัวแปร> = <ชื่อตัวอ่าน>.nextInt();
```

หลังชื่อเมทอด nextInt() ต้องตามด้วยวงเล็บเสมอห้ามลืมเด็ดขาด

เช่น int num = in.nextInt();

(2) คำสั่งอ่านค่าจำนวนจริงประเภท double โดยใช้เมทอด nextDouble()

```
double <ชื่อตัวแปร> = <ชื่อตัวอ่าน>.nextDouble();
```

เช่น double d = in.nextDouble();

(3) คำสั่งอ่านค่าข้อมูลประเภท String ที่ละบรรทัด โดยใช้เมทอด nextLine()

```
String <ชื่อตัวแปร> = <ชื่อตัวอ่าน>.nextLine();
```

เช่น String s1 = in.nextLine();

(4) คำสั่งอ่านค่าข้อมูลประเภท String ที่ละคำหรือที่ละช่อง (แต่ละช่องคั่นด้วยช่องว่างหรือ \t หรือ \n) โดยใช้เมทอด next()

```
String <ชื่อตัวแปร> = <ชื่อตัวอ่าน>.next();
```

เช่น String s2 = in.next();

- 6) การปิดแฟ้มข้อมูล เมื่ออ่านข้อมูลเสร็จสิ้นเรียบร้อยแล้ว จะต้องทำการปิดแฟ้มข้อมูลที่เราอ่านนั้นเสมอ โดยใช้เมทอด close() ซึ่งมีรูปแบบคำสั่งดังต่อไปนี้

```
<ชื่อตัวอ่าน>.close();
```

เช่น in.close();

- 7) ตัวอย่างการเขียนโปรแกรมเปิดอ่านแฟ้มข้อมูลแบบครบทุกขั้นตอน

```
1 import java.util.Scanner;
2 import java.io.*;
3 public class Test {
4     public static void main(String[] args) throws IOException {
5         Scanner in = new Scanner(new File("d:/cu/student.dat"));
6         int i = 1;
7         while(in.hasNext()) {
8             String s = in.nextLine();
9             System.out.println(i + " : " + s);
10            i++;
11        }
12        in.close();
13    }
14 }
```

ชื่อแฟ้มข้อมูลนี้เป็นการเรียกชื่อแบบ Full Path

เปิดอ่านแฟ้มข้อมูลที่ชื่อ student.dat

หมายเลขบรรทัด

วนอ่านข้อมูลที่ละบรรทัดจากแฟ้มข้อมูลแล้วแสดงผลลัพธ์ของแต่ละบรรทัดที่อ่านได้ขึ้นบนจอภาพ

ปิดแฟ้มข้อมูล

โจทย์ข้อที่ 1 [ระดับง่าย] จงแสดงผลลัพธ์จากส่วนของโปรแกรมต่อไปนี้ตามคำสั่งที่ระบุ (12 คะแนน)

```

1 Scanner in = new Scanner(new File("test.txt"));
2 while (in.hasNext()) {
3     System.out.println(.....);
4 }
5 in.close();

```

test.txt

```

1      2      3
4
5  6
7.0    8    9D 0

```

in.nextInt()	in.nextDouble()	in.nextLine()	in.next()

จำนวนรอบของ while	จำนวนรอบของ while	จำนวนรอบของ while	จำนวนรอบของ while

โจทย์ข้อที่ 2 [ระดับง่าย] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่ออ่านแฟ้มข้อมูลที่ชื่อ data.txt ซึ่งภายในเก็บข้อมูลที่เป็นข้อความจำนวนหนึ่งที่ไม่ทราบจำนวนบรรทัดที่แน่ชัด แล้วทำการนับจำนวนบรรทัด (Number of Lines) ทั้งหมดที่ปรากฏอยู่ในแฟ้มข้อมูลนั้น พร้อมทั้งแสดงผลลัพธ์ที่ได้ออกทางจอภาพ (10 คะแนน)

โจทย์ข้อที่ 3 [ระดับง่าย] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่ออ่านแฟ้มข้อมูลที่มีชื่อ data.txt ซึ่งภายในเก็บข้อมูลที่เป็นข้อความจำนวนหนึ่งที่ไม่ทราบจำนวนบรรทัดที่แน่ชัด แล้วทำการนับจำนวนคำ (Number of Words) ทั้งหมดที่ปรากฏอยู่ในแฟ้มข้อมูลนั้น พร้อมทั้งแสดงผลลัพธ์ที่ได้ออกทางจอภาพ (10 คะแนน)

โจทย์ข้อที่ 4 [ระดับง่าย] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่ออ่านแฟ้มข้อมูลที่มีชื่อ data.txt ซึ่งภายในเก็บข้อมูลที่เป็นข้อความจำนวนหนึ่งที่ไม่ทราบจำนวนบรรทัดที่แน่ชัด แล้วทำการนับจำนวนอักขระ (Number of Characters) ทั้งหมดที่ปรากฏอยู่ในแฟ้มข้อมูลนั้น โดยอักขระในที่นี้อาจจะเป็นตัวอักษร ตัวเลข หรือ สัญลักษณ์ต่างๆ ก็ได้ พร้อมทั้งแสดงผลลัพธ์ที่ได้ออกทางจอภาพ (10 คะแนน)

โจทย์ข้อที่ 5 [ระดับปานกลาง] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่ออ่านแฟ้มข้อมูล student.txt ซึ่งเป็นแฟ้มข้อมูลที่เก็บรายชื่อ นิสิตที่ลงทะเบียนเรียน รายวิชา 2110191 Innovative Thinking เพื่อตรวจนับ ว่ามี นิสิต คณะวิศวกรรมศาสตร์ ที่ลงทะเบียนในวิชานี้จำนวนกี่คน โดยภายในแฟ้มมีรูปแบบข้อมูลตามดังตัวอย่างข้างล่างนี้ (10 คะแนน)

student.txt				ตัวอย่างการแสดงผลบนจอภาพ
1.	Wongyos	Keardsri	5271822821	Engineering Students: 2
2.	Wannaporn	Yim-ngam	5330023422	
3.	Ong-ard	Rutsamee	5230586021	
...				

หมายเหตุ ตัวเลขสองหลักสุดท้ายของเลขประจำตัว นิสิต คือรหัสคณะ ซึ่งรหัสของคณะวิศวกรรมฯ คือ 21

```
import java.util.Scanner;
import java.io.*;
public class CountEngStudent {
    public static void main(String[] args) throws IOException {
```

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

โจทย์ข้อที่ 6 [ระดับปานกลาง] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่ออ่านแฟ้มข้อมูล score.txt ซึ่งประกอบด้วยเลขประจำตัว นิสิต คะแนนรวมสุทธิ (เต็ม 100 คะแนน) และรหัสคณะ เพื่อคำนวณหาผล การเรียนของนิสิตแต่ละคนและแสดงผลลัพธ์ออกทางจอภาพตามตัวอย่างต่อไปนี้ โดยถ้า นิสิตได้คะแนน ตั้งแต่ 60 คะแนนขึ้นไปจะได้ผลการเรียนเป็น “S” แต่ถ้าต่ำกว่านั้นได้ผลการเรียนเป็น “U” (10 คะแนน)

score.txt			ตัวอย่างการแสดงผลบนจอภาพ		
5330000221	45.23	21	1.	5330000221	U
5330000320	72.45	20	2.	5330000320	S
5330000421	91.11	21	3.	5330000421	S
5330000521	38.20	21	4.	5330000521	U
5330000622	60.00	22	5.	5330000622	S
...			...		

```
import java.util.Scanner;
import java.io.*;
public class StudentGrade {
    public static void main(String[] args) throws IOException {
```

```

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

```

โจทย์ข้อที่ 7 [ระดับยาก] ฝ่ายวัดผลและประเมินผลของสำนักทะเบียนจุฬาลงกรณ์มหาวิทยาลัยได้ส่งแฟ้มข้อมูลชื่อ student.dat มาให้กับฝ่ายพัฒนาโปรแกรมของบริษัทลานเกียร์จำกัด ซึ่งประกอบไปด้วยข้อมูลของนิสิตดังนี้ เลขประจำตัว ชื่อ นามสกุล และ ผลการเรียนเฉลี่ย โดยถูกบันทึกไว้ในรูปแบบตามที่ปรากฏ ซึ่งฝ่ายพัฒนาฯ จะต้องเขียนโปรแกรมภาษาจาวาเพื่ออ่านแฟ้มข้อมูลดังกล่าวแล้วนำมาประมวลผลดังรายการต่อไปนี้ (15 คะแนน)

- 1) คำนวณหาชั้นปีปัจจุบันของนิสิตแต่ละคน เช่น 5330000121 คือ 1st
- 2) เปลี่ยนชื่อนิสิตให้เหลือเพียง 1 ตัวอักษรย่อ โดยใช้ตัวอักษรตัวแรกของชื่อเต็มของนิสิตแต่ละคน เช่น Preecha จะได้เป็น P. หรือ wongyos จะได้เป็น W. โดยที่อักษรย่อต้องเป็นอักษรพิมพ์ใหญ่เท่านั้น
- 3) แสดงสถานะการเรียนของนิสิตแต่ละคน โดยถ้ามีผลการเรียนตั้งแต่ 2.00 ขึ้นไปให้แสดงสถานะว่า "Pass" ถ้าอยู่ระหว่าง 1.00–2.00 ให้แสดงสถานะว่า "Critical" ถ้าต่ำกว่า 1.00 ให้แสดงสถานะว่า "Retired"
- 4) แสดงข้อมูลของนิสิตแต่ละคนให้อยู่ภายในบรรทัดเดียวกันดังตัวอย่างต่อไปนี้

student.dat

```

4830000121
preecha
Lerdsirikul
1.99
5230000221
Suwaraporn
Chokkhokhang
2.02
5330000321
Ruksajai
Suksawatdee
0.56
...

```

```

1.      4830000121    6th    P. Lerdsirikul    1.99    Critical
2.      5230000221    2nd    S. Chokkhokhang  2.02    Pass
3.      5330000321    1st    R. Suksawatdee   0.56    Retired
...

```

```

import java.util.Scanner;
import java.io.*;
public class StudentInfoFromFile {
    public static void main(String[] args) throws IOException {

```

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

โจทย์ข้อที่ 8 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อนับคำว่า “love” จากแฟ้มข้อมูล `song.txt` โดยไม่สนใจอักขรพิมพ์ใหญ่และอักขรพิมพ์เล็ก ซึ่งกำหนดให้มีวิธีการนับ 2 วิธีคือ วิธีที่หนึ่ง นับแบบสนใจช่องว่างที่แบ่งคำ (นับทีละคำ) เช่น “love you my lovely” จะมีคำว่า “love” 1 คำ และวิธีที่สองนับแบบไม่สนใจช่องว่างที่แบ่งคำ (เอาช่องว่างออก) เช่น “love you my lovely” จะมีคำว่า “love” 2 คำ พร้อมทั้งแสดงผลลัพธ์ของการนับทั้งสองวิธีออกทางจอภาพดังตัวอย่างต่อไปนี้ (15 คะแนน)

song.txt

```
I Love you all over LOvE java.
I lOvE some lovely friend.
LovelOvE java!!.
```

ตัวอย่างการแสดงผลบนจอภาพ

```
Count Words #1: 3
Count Words #2: 7
```


โจทย์ข้อที่ 9 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่ออ่านแฟ้มข้อมูล data1.dat และ data2.dat ซึ่งเก็บข้อมูลผลการเรียนของนิสิตที่ลงทะเบียนรายวิชา 2110101 ภาคการศึกษาต้น ของปีการศึกษา 2552 และปีการศึกษา 2553 ตามลำดับ จากนั้นให้คำนวณหาจำนวนนิสิตที่ได้ผลการเรียนเป็น “F” ว่ามีร้อยละเท่าไรของนิสิตที่ลงทะเบียนในภาคการศึกษานั้นๆ และแสดงผลว่าปีการศึกษาใดที่มีนิสิตได้ผลการเรียนเป็น “F” มากกว่ากัน (15 คะแนน)

data1.dat				data2.dat			
No.	Student ID	Grade		No.	Student ID	Grade	
=====				=====			
1.	5230000121	C+		1.	5330000121	F	
2.	5230000221	A		2.	5330000221	F	
3.	5230000321	B+		3.	5330000321	F	
4.	5230000421	F		4.	5330000421	D+	
5.	5230000521	B		5.	5330000521	A	
...				...			

```
import java.util.Scanner;
import java.io.*;
public class FComparison {
    public static void main(String[] args) throws IOException {
```

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

2. ขั้นตอนการเขียนแฟ้มข้อมูล

การเขียนแฟ้มข้อมูลในภาษาจาวาสามารถทำได้หลายวิธีเช่นเดียวกับการอ่านแฟ้มข้อมูล แต่ในบทนี้จะนำเสนอการเขียนแฟ้มข้อมูลด้วยคลาส `PrintStream` โดยมีขั้นตอนต่างๆ ดังต่อไปนี้

- 1) การนำเข้าคลาส `PrintStream` ก่อนการเปิดเขียนแฟ้มข้อมูลจะต้องนำเข้าคลาส `PrintStream` ซึ่งอยู่ในแพ็คเกจ `java.io` ก่อนเสมอ โดยใช้คำสั่ง `import java.io.*;` ซึ่งระบุไว้ก่อนเขียนหัวคลาส โดยปฏิบัติเช่นเดียวกับการเปิดอ่านแฟ้มข้อมูล
- 2) การตรวจสอบความผิดพลาดในการเปิดเขียนแฟ้มข้อมูล เมธอดใดก็ตามที่เป็นเมธอดสำหรับเปิดเขียนแฟ้มข้อมูล จะต้องมีการระบุส่วนของคำสั่ง `throws IOException` ไว้ตอนท้ายของหัวเมธอดนั้นเสมอ ซึ่งปฏิบัติเช่นเดียวกับการเปิดอ่านแฟ้มข้อมูล เพื่อให้ระบบตรวจสอบข้อผิดพลาด

- 3) การสร้างตัวเขียนจากแฟ้มข้อมูล เป็นการกำหนดชื่อตัวเขียนเพื่อใช้สำหรับเขียนข้อมูลต่างๆ ลงไปในแฟ้มข้อมูล โดยจะสร้างเพียงครั้งเดียวเท่านั้น และจะใช้งานตัวเขียนนั้นตลอดทั้งโปรแกรม ซึ่งมีคำสั่งดังนี้

```
PrintStream <ชื่อตัวเขียน> = new PrintStream(new File("<ชื่อแฟ้มข้อมูล>"));
```

```
1 import java.util.Scanner;
2 import java.io.*;
3 public class Test {
4     public static void main(String[] args) throws IOException {
5         PrintStream out = new PrintStream(new File("data.txt"));
6         ...
7     }
8 }
```

ตัวเขียนแฟ้มข้อมูลชื่อว่า out

เปิดเขียนแฟ้มข้อมูลชื่อ data.txt

- 4) การเขียนข้อมูลด้วยตัวเขียนแฟ้มข้อมูล ตัวเขียนแฟ้มข้อมูลที่เราสร้างขึ้นสามารถเขียนข้อมูลประเภทจำนวนเต็ม จำนวนจริง และสตริง โดยใช้คำสั่งและเมทอดต่อไปนี้

- 1) คำสั่งเพื่อเขียนข้อมูลลงในแฟ้มข้อมูลทีละบรรทัดแต่ไม่มีการขึ้นบรรทัดใหม่

```
<ชื่อตัวเขียน>.print(<ข้อมูล>);
```

ทำงานคล้ายกับคำสั่ง
System.out.print(...);

เช่น out.print("Start New File");

- 2) คำสั่งเพื่อเขียนข้อมูลลงในแฟ้มข้อมูลทีละบรรทัดพร้อมทั้งขึ้นบรรทัดใหม่

```
<ชื่อตัวเขียน>.println(<ข้อมูล>);
```

ทำงานคล้ายกับคำสั่ง
System.out.println(...);

เช่น out.println("Start New File");

- 5) การปิดแฟ้มข้อมูล เมื่อเขียนข้อมูลเสร็จสิ้นเรียบร้อยแล้ว จะต้องทำการปิดแฟ้มข้อมูลที่เขียนนั้นเสมอ โดยใช้เมทอด close() ซึ่งปฏิบัติเช่นเดียวกับการอ่านแฟ้มข้อมูล

- 6) ตัวอย่างการเขียนโปรแกรมเปิดเขียนแฟ้มข้อมูลแบบครบทุกขั้นตอน

```
1 import java.util.Scanner;
2 import java.io.*;
3 public class Test {
4     public static void main(String[] args) throws IOException {
5         PrintStream out = new PrintStream(new File("data.dat"));
6         for(int i = 1; i <= 10; i++) {
7             out.println("Line #" + i);
8         }
9         out.close();
10    }
11 }
```

ถ้าต้องการรับค่าจากแป้นพิมพ์ ก็ต้องมีคำสั่งนี้ด้วย

ปิดแฟ้มข้อมูล

วนเขียน (บันทึก) ข้อมูลลงในแฟ้มข้อมูลที่ละบรรทัด โดยจะไม่มีผลลัพธ์แสดงที่จอภาพ แต่ผลลัพธ์จะแสดงอยู่ภายในแฟ้มข้อมูลแทน

โจทย์ข้อที่ 10 [ระดับง่าย] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อรับประโยคจากแป้นพิมพ์ทีละประโยค แล้วบันทึกลงในแฟ้มข้อมูลที่ชื่อ sentence.txt โดยไม่ต้องแสดงผลที่จอภาพ ซึ่งจะวนรับประโยคไปเรื่อยๆ จนกว่าจะพิมพ์ประโยคว่า "Stop" ก็จะจบการวนซ้ำ (10 คะแนน)

โจทย์ข้อที่ 11 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่ออ่านแฟ้มข้อมูล `text.txt` แล้วทำการกลับข้อความหรือประโยคที่ปรากฏอยู่ในแฟ้มข้อมูลที่ละบรรทัด เช่น `we love java` จะได้เป็น `avaj evol ew` เป็นต้น พร้อมทั้งเก็บผลลัพธ์ที่ได้ไว้ในแฟ้มข้อมูล `revtext.txt` (15 คะแนน)

`text.txt`

```
We love java
Java is my first programming and
Java is my last programming
...
```

`revtext.txt`

```
avaj evol ew
dna gnimmargorp tsrif ym si avaJ
gnimmargorp tsal ym si avaJ
...
```