

2110101 : การทำโปรแกรมคอมพิวเตอร์ ระบบคอมพิวเตอร์เบื้องต้น

ภาควิชา工ศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

คอมพิวเตอร์

- เครื่องประมวลผล จัดเก็บ และสื่อสารข้อมูล
 - ประมวลผลข้อมูลได้รวดเร็ว
 - จัดเก็บข้อมูลได้มากมาย
 - สื่อสารกันได้อย่างไว้พร้อมแน่น
- การใช้งาน
 - บันเทิง
 - เกมส์ ดูหนัง พิงเพลง ...
 - จัดเก็บและค้นคืน
 - WWW, ระบบทะเบียนราชภัฏ ระบบห้องสมุด ...
 - ธุรกิจ
 - email ระบบบุคลากร, วิเคราะห์ข้อมูล, วางแผนทรัพยากร ...
 - ...



วัตถุประสงค์ของ 2110101

- รู้จักคอมพิวเตอร์
 - ประยุกต์คอมพิวเตอร์
 - แก้ปัญหาทางวิศวกรรมด้วยคอมพิวเตอร์
- ทำโปรแกรมคอมพิวเตอร์

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 2

คุณลักษณะ

- ชื่อสัตย์ ทำตามคำสั่ง
- ไม่บ่น ทำแต่เรื่องซ้ำๆ ซากๆ
- จำได้มาก จำได้แม่น
- ทำงานรวดเร็ว
- ไม่ฉลาด สอนอย่างไรก็ทำอย่างนั้น



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

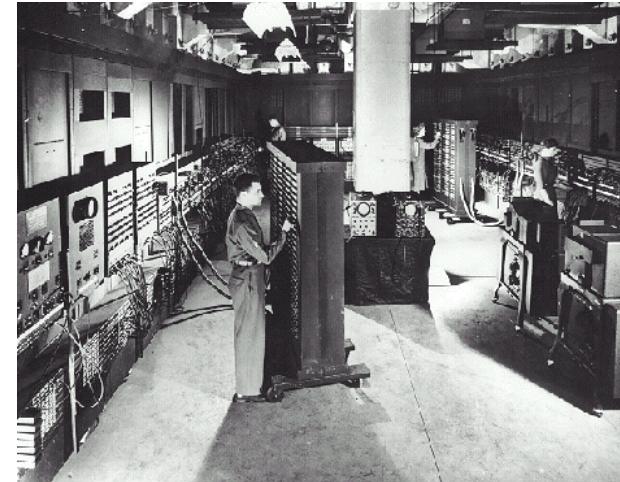
(20/5/02) 4

เราเพิ่งมีคอมพิวเตอร์ใช้กันแค่ 60 กว่าปี

- เล็กลง (ร้อยล้านทรานซิสเตอร์ป้ายนิวก้อย)
- เร็วขึ้น (พันล้านคำสั่งในวินาที)
- จุข้อมูลมากขึ้น (แสนล้านตัวอักษรในแฟ้มีอ)
- กินไฟน้อยลง (ใช้ไฟหั่งคำเฉลี่อเท่าทีวี)
- ราคาถูกลง (ใบตั๋วบาทเป็นแสนใบตั๋วบาท)

Moore's Law : เทคโนโลยีทางคอมพิวเตอร์มี
ประสิทธิภาพมากขึ้นสองเท่าทุกๆ 18 เดือน

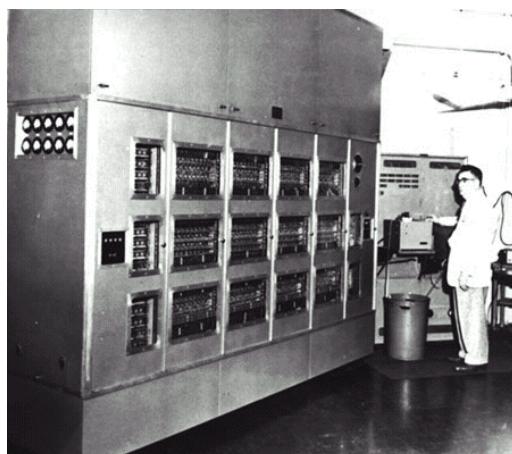
ENIAC (1946)



1946 : ENIAC เดิมใช้ห้องขนาด 30x50 ตารางฟุต

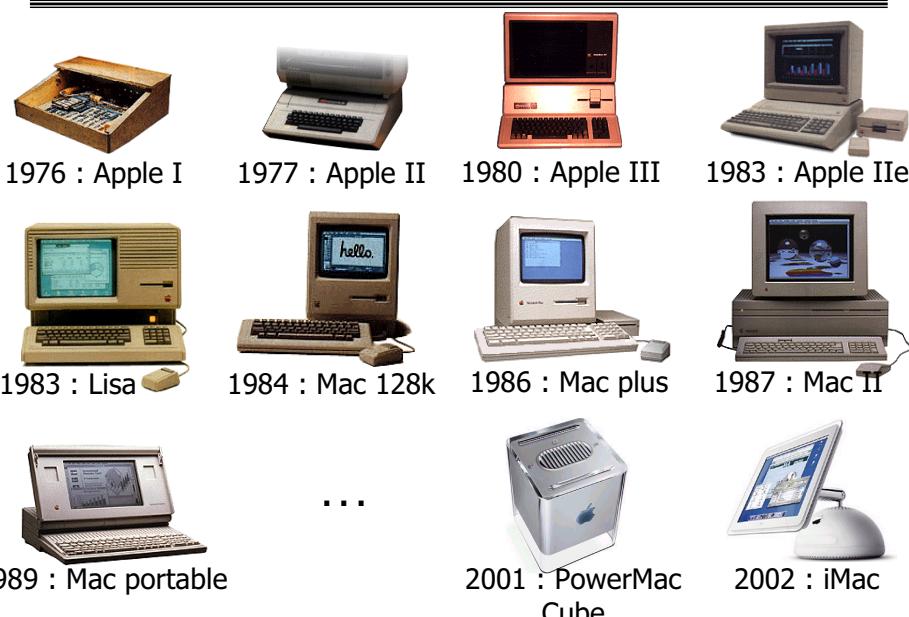
1996 : ENIAC chip ขนาด 7.44 x 5.29 ตาราง มม.

ILLIAC I (1952)



ใช้ 2,800 หลอด กว้างxยาวxสูง = $2 \times 10 \times 8.5$ ฟุต³
หนัก 5 ตัน มี memory 5.12K และ magnetic drum
64K ให้บริการเป็นเวลา 10 ปี

Apple Computers



องค์ประกอบของคอมพิวเตอร์

- hardware เพื่อประมวลผล จัดเก็บ และสื่อสารข้อมูล
- software เพื่อสั่งให้ hardware ทำงานตามขั้นตอนที่กำหนดไว้

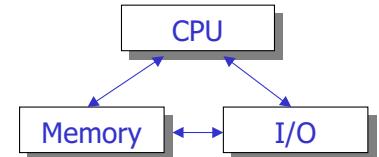


2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 9

Hardware

- Central Processing Unit (CPU)
 - เพื่อประมวลผลทางคณิตศาสตร์, ตรรกะ, การทดสอบจริงเท็จ, การทำงานซ้ำ
- Memory
 - เพื่อจัดเก็บและค้นคืนข้อมูล
- I/O
 - เพื่อรับส่งสื่อสารข้อมูล (จอภาพ แป้นพิมพ์ เม้าส์ โมเด็ม เครื่องพิมพ์ กล้อง ...)



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 10

Central Processing Unit

- ทำงานตามชุดคำสั่งที่กำหนดให้
- ทำงานได้รวดเร็ว (พันล้านคำสั่งในวินาที)
- ควบคุมการทำงานของส่วนอื่นๆ
- ชุดคำสั่ง
 - เป็นรหัสฐานสอง
 - เก็บในหน่วยความจำ
 - เปลี่ยนแปลงได้
- ยี่ห้อ
 - Intel Pentium, Intel Celeron, Intel StrongARM, AMD Xeon, SunSparce, IBM PowerPC, ...

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 11

CPU

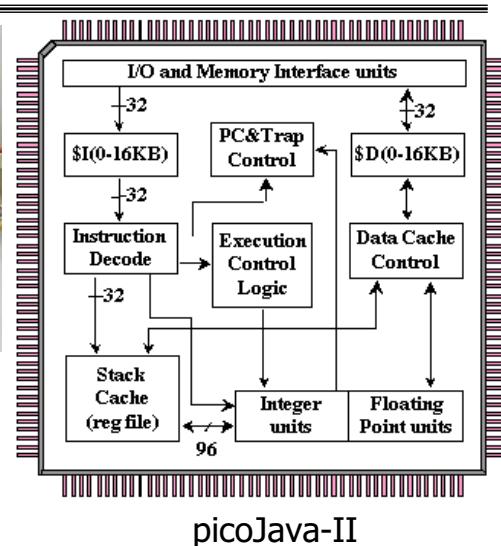


microprocessor

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 12

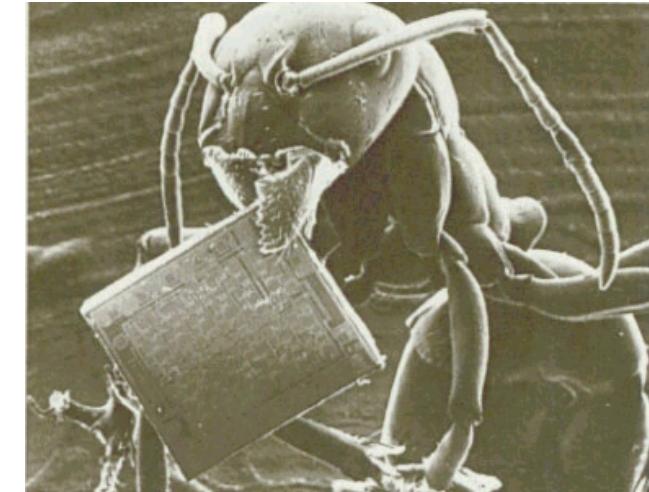
CPU



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 13

มดกับชิป



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 14

Memory

- เก็บชุดคำสั่งการทำงานของ CPU
- เก็บข้อมูลสำหรับการประมวลผล
- แบ่งเป็น
 - main memory
 - ที่เก็บชั่วคราว
 - เร็วแต่ราคาแพง (10^{-8} วินาที แสนใบต่อบาท)
 - RAM (random access memory)
 - secondary storage
 - ที่เก็บถาวร
 - ช้ากว่าแต่ราคาถูก (HD : 0.01 วินาที สิบล้านใบต่อบาท)
 - hard disk, CD-ROM, DVD-ROM, memory stick, ...

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 15

Memory



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 16

Input/Output Devices

- อุปกรณ์รับข้อมูล

- แป้นพิมพ์ เม้าส์ แป้นกดเกมส์ กล้อง สแกนเนอร์ โน้ตบุ๊ก
ไมโครโฟน เครื่องอ่านบาร์โค้ด เครื่องอ่านบัตร

- อุปกรณ์ส่งหรือแสดงข้อมูล

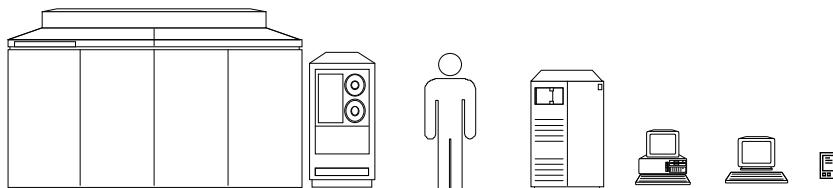
- จอภาพ เครื่องพิมพ์ โน้ตบุ๊ก ลำโพง



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 17

ประเภท : ขนาดและหน้าที่



- Mainframe
- Mini-computer
- Personal Computer
- Network Computer
- Personal Digital Assistant



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 19

หน่วยต่างๆ ที่นำเสนอ

- bit - หน่วยเล็กสุดแทนสองสภาวะ (binary digit)
- byte - 8 bits
- word - จำนวน bytes ที่เครื่องประมวลผลได้สะท้อน
- CPU : 900MHz, 2.5GHz
- RAM : 256MB
- Harddisk : 80GB, 7200rpm
- Printer : 600dpi, 10ppm
- Sound : 44.1KHz 16-bit stereo
- Modem : 56Kbps

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 18

Reliability

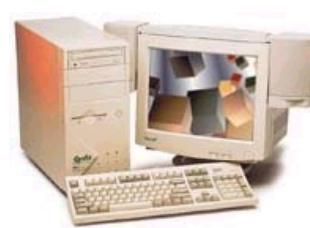
Type	Platform	Outages/year
Mainframe	IBM S/390 (Sysplexed)	10 min.
	Tandem Nonstop	1.7 Hrs.
Mini	Digital VAX	18.9 Hrs.
	Unix-based	23.8 Hrs.
PC	Windows-NT based	224.5 Hrs.

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 20

คอมพิวเตอร์ราคาถูก (NECTEC)

- CPU : 900MHz
- Memory
 - SDRAM, 64MB (133MHz)
 - HD Ultra DMA (100MB/s) 20GB
 - Diskette drive : 1.44MB
 - CD-ROM : 50X
- I/O
 - VGA card : AGP, 16MB
 - Monitor : 15", 85Hz at 800x600, 0.28mm
 - Sound card : PCI, Audio Codec 97
 - LAB card : TX10/100
 - Modem : V.90 56Kb/s



20,000 บาท

Software

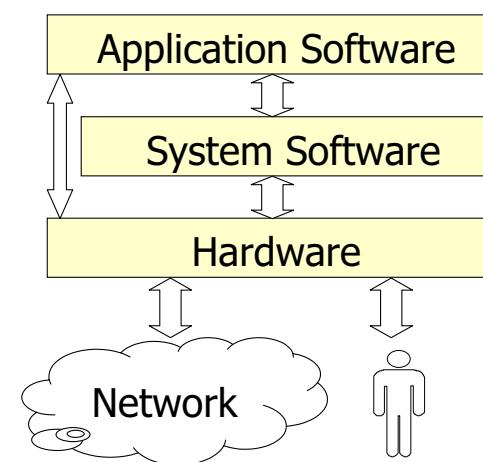
- ชุดคำสั่งต่างๆ ซึ่งสั่งให้ hardware ทำงานตามขั้นตอนที่กำหนดไว้
- เราเก็บ software มากรามายไว้ใน secondary storage (harddisk, CD-ROM, DVD-ROM)
- ระบบนำ (หรือ load) software ที่ต้องการลงใน main memory เพื่อใช้งาน



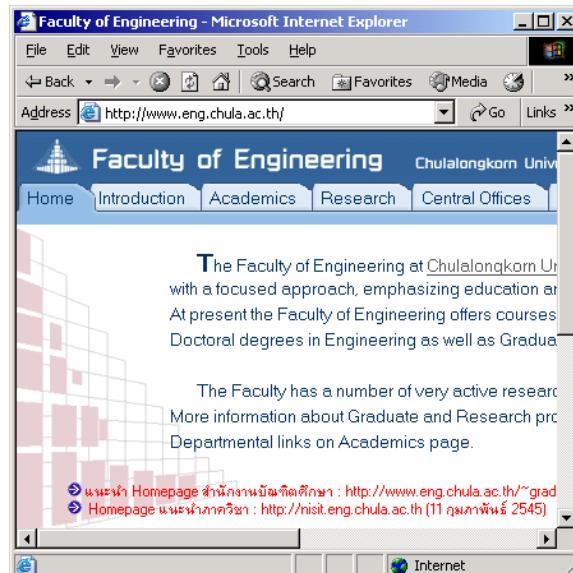
Software

- System software
 - Operating system (Windows XP, Linux, Mac OS, ...)
 - System tools (antivirus, zip/unzip, backup/restore, ...)
 - Development tools (compiler, debugger, ...)
 - ...
- Application software
 - Office (email, word processor, spreadsheet, ...)
 - Network applications (browser, chat, ...)
 - Entertainment (movies, musics, games, ...)
 - ...

Hardware & Software



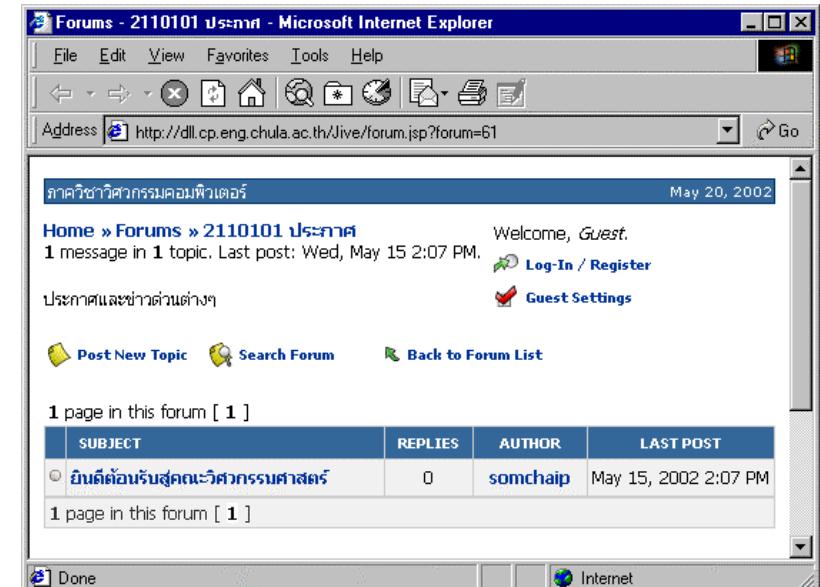
Web Browser



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 25

Web Board

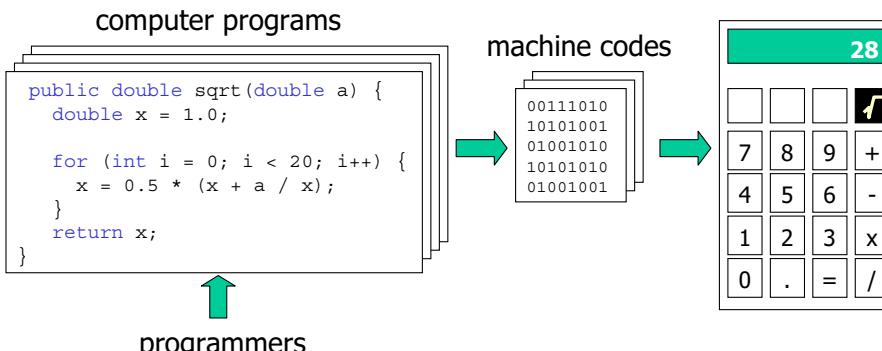


2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 26

Software

- เราพัฒนา software ได้ด้วยการเขียนโปรแกรมคอมพิวเตอร์ (computer program)
 - โปรแกรมคือชุดคำสั่งซึ่งเขียนด้วยภาษาคอมพิวเตอร์ เพื่อสั่งงานคอมพิวเตอร์ให้มีการตามข้อกำหนด



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 27

ขั้นตอนการพัฒนา software

- กำหนดความต้องการ (ปัญหา)
- วิเคราะห์ตัวปัญหา
- ออกแบบขั้นตอนวิธีแก้ไขปัญหา
- ทำโปรแกรมตามขั้นตอนวิธีที่ออกแบบ
- ทดสอบความถูกต้อง
- ปรับปรุงแก้ไขโปรแกรม

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

(20/5/02) 28

ตัวอย่าง

- ความต้องการ
 - ค่าเฉลี่ยของชุดข้อมูล
- วิเคราะห์
 - แหล่งข้อมูล
 - ผู้ใช้ป้อนจำนวนข้อมูล
 - ผู้ใช้ป้อนข้อมูลต่างๆ ผ่านทางแป้นพิมพ์
 - ผลลัพธ์
 - แสดงค่าเฉลี่ยออกสู่จอภาพ

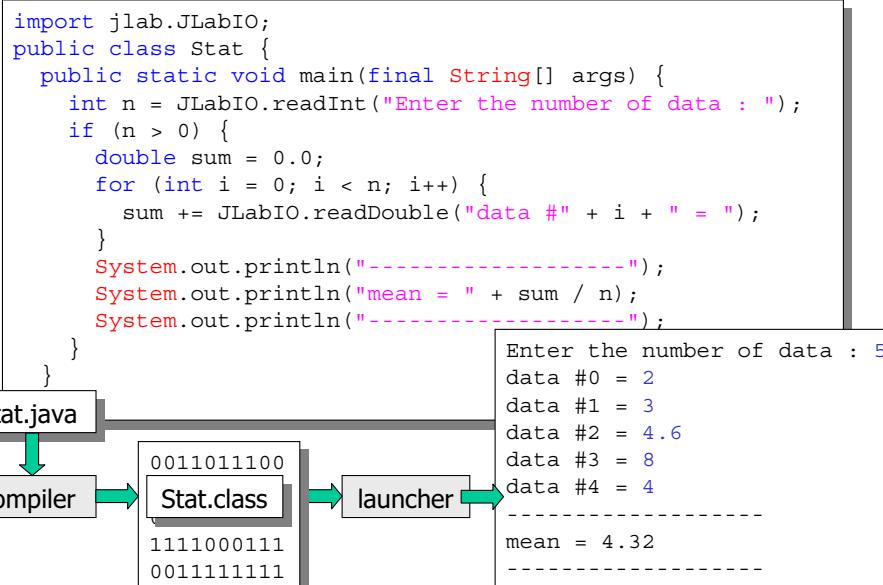
ตัวอย่าง

- ออกแบบ
 - รับจำนวนข้อมูล n จากผู้ใช้
 - รับข้อมูลต่างๆ จากผู้ใช้ x_1, x_2, \dots, x_n
 - ผลลัพธ์คำนวณได้จากสูตร

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- แสดงผลลัพธ์ออกสู่จอภาพ

ตัวอย่าง



การทำโปรแกรม

- edit : เขียนโปรแกรมด้วยภาษาคอมพิวเตอร์
- compile : แปลงโปรแกรมที่เขียนให้เป็นรหัสเครื่อง
- execute : สั่งงานรหัสเครื่องที่ได้
- debug : หาที่ผิดพลาดของโปรแกรม
- เรามักใช้ Integrated programming environment software เป็นเครื่องมือในการพัฒนาซอฟต์แวร์แบบเบ็ดเสร็จสำหรับวงจรการ “edit-compile-execute-debug”

JLab

The screenshot shows the JLab 3.06.01 IDE interface. The top window is titled "JLab 3.06.01 [Statistics]". The menu bar includes File, Edit, Build, Debug, Tools, Window, and Help. The code editor displays the following Java code:

```
1 import jlab.JLabIO;
2 public class Stat {
3     public static void main(final String[] args) {
4         int n = JLabIO.readInt("Enter the number of data");
5         if (n > 0) {
6             double sum = 0.0;
7             for (int i = 0; i < n; i++) {
8                 sum += JLabIO.readDouble("data #" + i + " = ");
9             }
10            System.out.println("-----");
11            System.out.println("mean = " + sum / n);
12            System.out.println("-----");
13        }
14    }
15 }
```

The status bar at the bottom shows "JLab>java Selftest" and "mean = 12.05". The terminal window below the code editor shows the output of the program:

```
JLab>java Selftest
Enter the number of data : 4
data #0 = 12.2
data #1 = 13.4
data #2 = 12.6
data #3 = 10.0
-----
mean = 12.05
```

2110101 : การทำโปรแกรมคอมพิวเตอร์

แนวคิดการทำโปรแกรม

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อ

- องค์ประกอบของโปรแกรม
- ขั้นตอนการแปลและสั่งงานโปรแกรม
- การเขียนโปรแกรมให้คนและเครื่องเข้าใจ
- การทำงานแบบลำดับ
- ขั้นตอนการทำโปรแกรม

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 2

โปรแกรมแรก

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- โปรแกรมประกอบด้วยคำต่างๆ นำมาเขียนให้ถูกต้องตามไวยากรณ์
- คำต่างๆ ถูกคั่นด้วยช่องว่าง ตัวอักษรพิเศษ หรือการขึ้นบรรทัดใหม่
- คำต่างๆ มีทั้งคำส่วนของภาษา คำที่ระบบใช้ คำที่ตั้งขึ้นใหม่ และอื่นๆ

โปรแกรมประกอบด้วย class

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- โปรแกรมนี้ประกอบด้วย class หนึ่ง class ชื่อ Hello1
- รายละเอียดของ class Hello1 อยู่ภายในวงเล็บ { } หลังข้อความ
`public class Hello1`

Class ประกอบด้วย method

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- method บรรยายขั้นตอนการทำงาน
- รายละเอียดของ method อยู่ภายในวงเล็บ {} หลัง หัว method

Method ประกอบด้วยชุดคำสั่ง

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- แต่ละบรรทัดคำสั่งปิดท้ายด้วย ; (semicolon)
- System.out.print แสดงข้อความภายในวงเล็บออกทางจอภาพ
- println ทำงานเหมือน print แต่ขึ้นบรรทัดใหม่ด้วย

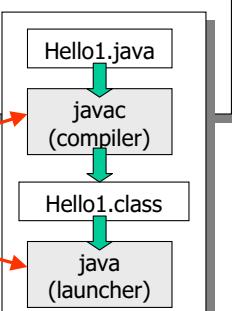
การแปลและสั่งทำงาน

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

```
JLab>javac Hello1.java
JLab>java Hello1
Hello 2110101
JLab>
```



main method

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- java launcher จะเรียกใช้ main method ของ class
- ดังนั้นการทำงานจะเริ่มที่ main method ของ class นั้น

Comment

```
// 2110101 : Computer Programming  
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- Comment เริ่มด้วย // (ยังมีวิธีเขียนแบบอื่น)
- โปรแกรมเมอร์เขียน comment ให้คนอ่าน เครื่องไม่สนใจ
- ตัด comment ทิ้ง โปรแกรมก็ยังทำงานได้เหมือนเดิม

สวยเกินไปอาจผิดกฎหมาย

```
*****  
* p u b l i c   c l a s s   H e l l o 1 {  
*  
*     P U B L I C   S T A T I C   V O I D   m a i n ( f i n a l   S t r i n g [ ]   a r g s ) {  
*  
*         S y s t e m . o u t . p r i n t ( " H e l l o " );  
*  
*         S y s t e m . o u t . p r i n t l n ( " 2 1 1 0 1 0 1 " );  
*  
*     }  
*  
* }  
*****
```

การเขียนโปรแกรมให้อ่านง่ายๆ

```
public class Hello1{public static void  
main(final String[] args){System.out.print("Hello ");  
System.out.println  
    ("2110101");  
}}
```

สองโปรแกรมนี้ทำงานเหมือนกัน
แต่โปรแกรมไหนอ่านง่ายกว่ากัน ?

```
public class Hello1 {  
    public static void main(final String[] args) {  
        System.out.print("Hello ");  
        System.out.println("2110101");  
    }  
}
```

- หน้าใช้ช่องว่าง การย่อหน้า และการขั้นบรรทัดใหม่ เพื่อให้โปรแกรมน่าอ่าน และอ่านง่ายขึ้น

การใช้สีเพื่อให้อ่านง่ายขึ้นอีก

- สีเขียว : comment
- สีน้ำเงิน : คำส่วนของภาษาจาวา
- สีแดง : คำที่พบในคลังคำสั่งมาตรฐานของจาวา
- สีส้ม : อักขระที่อยู่ภายใต้เครื่องหมาย "
- สีม่วง : อักขระที่อยู่ภายใต้เครื่องหมาย '
- สีดำ : อื่นๆ

```
// 2110101 : Computer Programming  
public class Hello1 {  
    public static void main(final String[] args) {  
        System.out.print("Hello ");  
        System.out.print("2110101");  
        System.out.print('A');  
    }  
}
```

การใช้สีเพื่อสังเกตความผิดพลาด

```
// 2110101 : Computer Programming
public Class Hello1 {
    public static void main(final String[] args) {
        System.out.print("Hello");
        System.out.print("2110101");
        System.out.print('A');
    }
}
```

- `Class` ต้องเป็น `class` (ต้องเป็น C ตัวเล็ก)
- `final` ต้องเป็น `final` (ต้องเป็นตัวแอลเล็ก 'ไม่ใช่เลข 1')
- `"Hello";` ต้องเป็น `"Hello")`; (ตัว " ปิดท้ายสายไป)
- `System` ต้องเป็น `System` (ต้องเป็น m)

การใส่สีเป็นความสามารถของส่วนบุồnโปรแกรม
เพื่อค้นอ่าน ตัวแปลลักษานั้นไม่สนใจสีเลย

การแสดงแบบวินโดว์

```
import javax.swing.JOptionPane;

public class HelloDialog {
    public static void main(final String[] args) {
        JOptionPane.showMessageDialog(null, "Hello");
        System.exit(0);
    }
}
```



- เราใช้คลัง `class` มาตรฐานของ Java `javax.swing.JOptionPane`
- บรรทัด `import` บอกว่าโปรแกรมนี้ต้องการใช้ `JOptionPane`
- ต้องเติม `System.exit(0)` เพื่อให้โปรแกรมหยุดทำงาน

บรรทัดคำสั่งทำงานจากบนลงล่าง

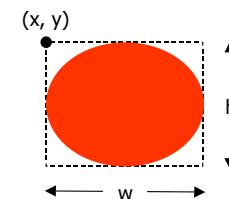
- โดยทั่วไปโปรแกรมจะทำงานเป็นลำดับ จากบรรทัดคำสั่งบนลงล่าง

```
public class PrintStars {
    public static void main(final String[] args) {
        System.out.println("*****");
        System.out.println("*      *");
        System.out.println("*      *");
        System.out.println("*      *");
        System.out.println("*****");
    }
}
```

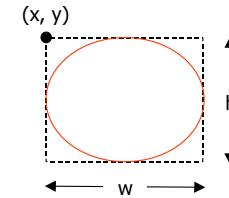
```
JLab>javac PrintStars.java
JLab>java PrintStars
*****
*      *
*      *
*      *
*****
JLab>
```

ตัวอย่างคำสั่งวาดรูป

```
g.setColor(Color.red);
g.fillOval(x, y, w, h);
```

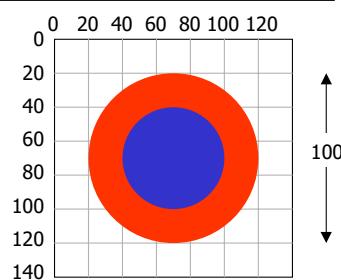


```
g.setColor(Color.red);
g.drawOval(x, y, w, h);
```

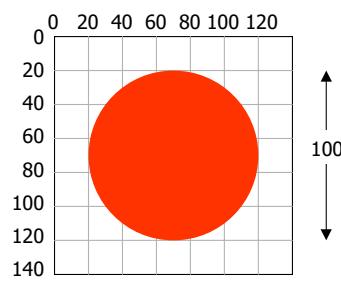


พิกัดบนจอภาพ

```
public void paint(Graphics g) {  
    g.setColor(Color.red);  
    g.fillOval(20, 20, 100, 100);  
    g.setColor(Color.blue);  
    g.fillOval(40, 40, 60, 60);  
}
```



```
public void paint(Graphics g) {  
    g.setColor(Color.blue);  
    g.fillOval(40, 40, 60, 60);  
    g.setColor(Color.red);  
    g.fillOval(20, 20, 100, 100);  
}
```



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 17

ตัวอย่างโปรแกรมวาดรูป

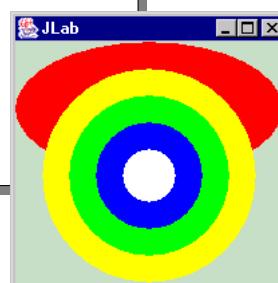
```
import java.awt.*;  
import java.awt.event.*;  
import java.applet.*;  
  
public class Monster extends Applet {  
  
    // main method  
    public static void main(String[] args) {  
        Frame frame = new Frame("JLab");  
        frame.addWindowListener(new WindowAdapter() {  
            public void windowClosing(WindowEvent e) {  
                System.exit(0);  
            }  
        });  
        frame.add(new Monster(), BorderLayout.CENTER);  
        frame.setSize(210, 210);  
        frame.setVisible(true);  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 18

ตัวอย่างโปรแกรมวาดรูป (ต่อ)

```
public void paint(Graphics g) {  
    g.setColor(Color.red);  
    g.fillOval(0, 0, 200, 100);  
    g.setColor(Color.yellow);  
    g.fillOval(20, 20, 160, 160);  
    g.setColor(Color.green);  
    g.fillOval(40, 40, 120, 120);  
    g.setColor(Color.blue);  
    g.fillOval(60, 60, 80, 80);  
    g.setColor(Color.white);  
    g.fillOval(80, 80, 40, 40);  
}  
} // end of Monster class
```



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 19

โปรแกรมหาผลบวก 1 ถึง 10

```
public class Add1to10 {  
    public static void main(String [] args) {  
        System.out.println(55);  
    }  
}  
  
public class Add1to10 {  
    public static void main(String [] args) {  
        System.out.println(1+2+3+4+5+6+7+8+9+10);  
    }  
}  
  
public class Add1to10 {  
    public static void main(String [] args) {  
        double x = 1+2+3+4+5+6+7+8+9+10;  
        System.out.println(x);  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 20

โปรแกรมหาค่าเฉลี่ยของสี่จำนวน

```
import jlab.JLabIO;

public class Mean {
    public static void main(String [] args) {
        double x = 0.0;
        x += JLabIO.readDouble("data #1 : ");
        x += JLabIO.readDouble("data #2 : ");
        x += JLabIO.readDouble("data #3 : ");
        x += JLabIO.readDouble("data #4 : ");
        System.out.println("mean = " + x / 4.0 );
    }
}
```

```
JLab>javac Mean.java
JLab>java Mean
data #1 : 10
data #2 : 12
data #3 : 14
data #4 : 14
mean = 12.5
JLab>
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 21

การทำโปรแกรม

- edit : เขียนโปรแกรมด้วยภาษาคอมพิวเตอร์
- compile : แปลงโปรแกรมที่เขียนให้เป็นรหัสเครื่อง
- execute : สั่งงานรหัสเครื่องที่ได้
- เรามักใช้ Integrated programming environment software เป็นเครื่องมือในการพัฒนาซอฟต์แวร์แบบเบ็ดเสร็จสำหรับวงการ “edit-compile-execute-debug”
- สำหรับวิชานี้เราจะใช้ JLab เป็นซอฟต์แวร์ในการช่วยพัฒนาโปรแกรม

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 22

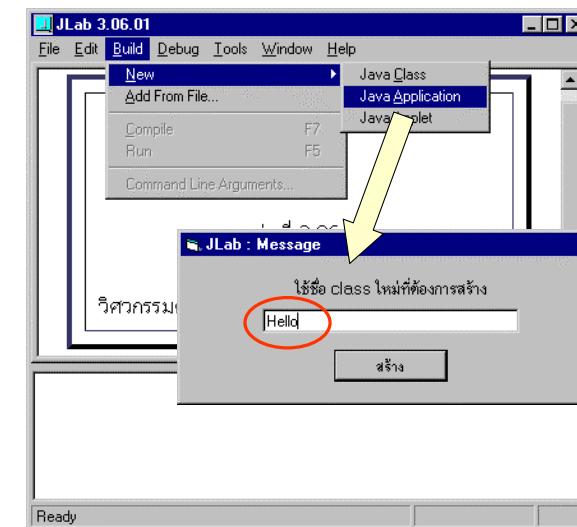
JLab



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 23

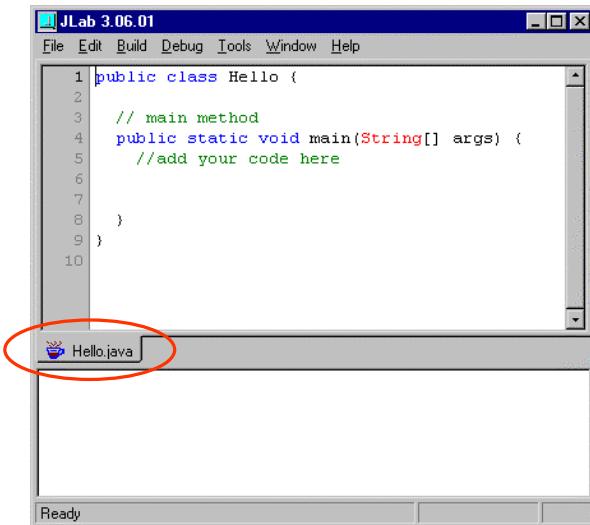
JLab : การสร้างแฟ้มโปรแกรม



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 24

JLab : การสร้างแฟ้มโปรแกรม

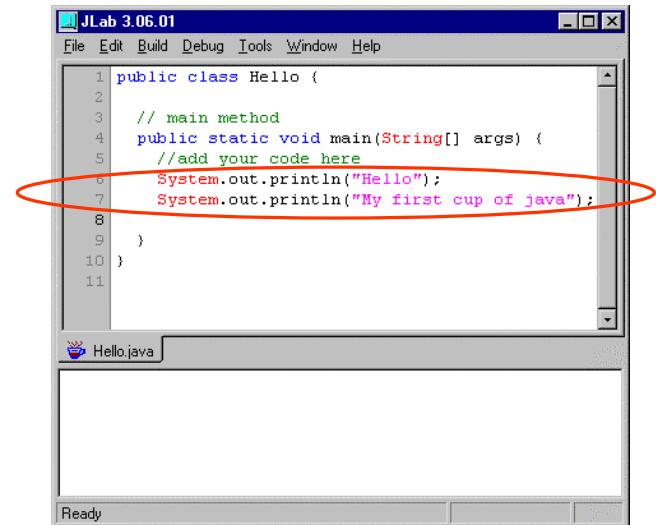


```
public class Hello {  
    // main method  
    public static void main(String[] args) {  
        //add your code here  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 25

JLab : การเขียนโปรแกรม

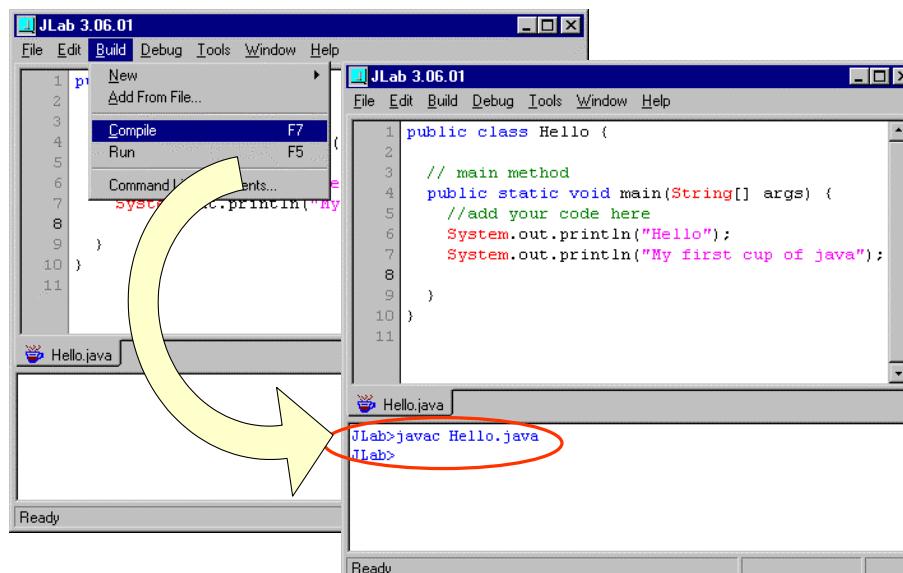


```
public class Hello {  
    // main method  
    public static void main(String[] args) {  
        //add your code here  
        System.out.println("Hello");  
        System.out.println("My first cup of java");  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 26

JLab : การแปลงโปรแกรม

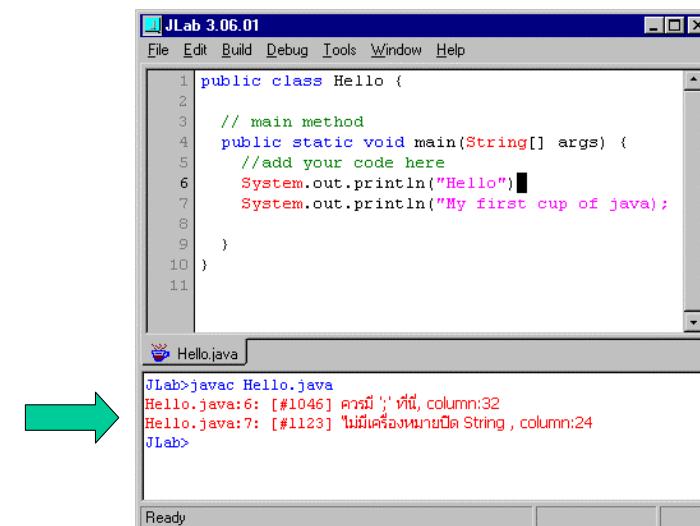


```
public class Hello {  
    // main method  
    public static void main(String[] args) {  
        //add your code here  
        System.out.println("Hello");  
        System.out.println("My first cup of java");  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 27

JLab : โปรแกรมที่ผิดกฎหมาย



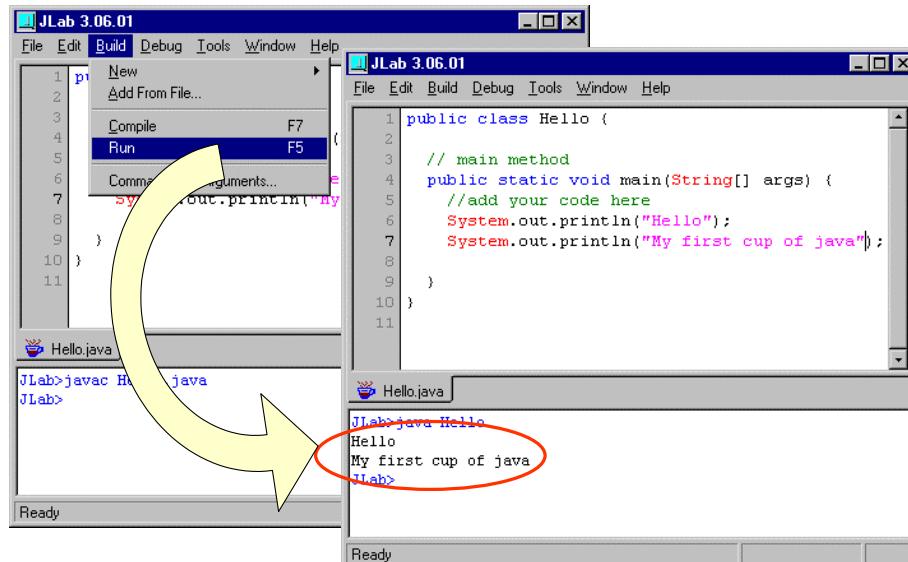
```
public class Hello {  
    // main method  
    public static void main(String[] args) {  
        //add your code here  
        System.out.println("Hello");  
        System.out.println("My first cup of java");  
    }  
}
```

JLab>javac Hello.java
Hello.java:6: [#1046] คาดว่า ';' ที่นี่, column:32
Hello.java:7: [##123] ไม่มีเครื่องหมาย界定 String , column:24
JLab>

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 28

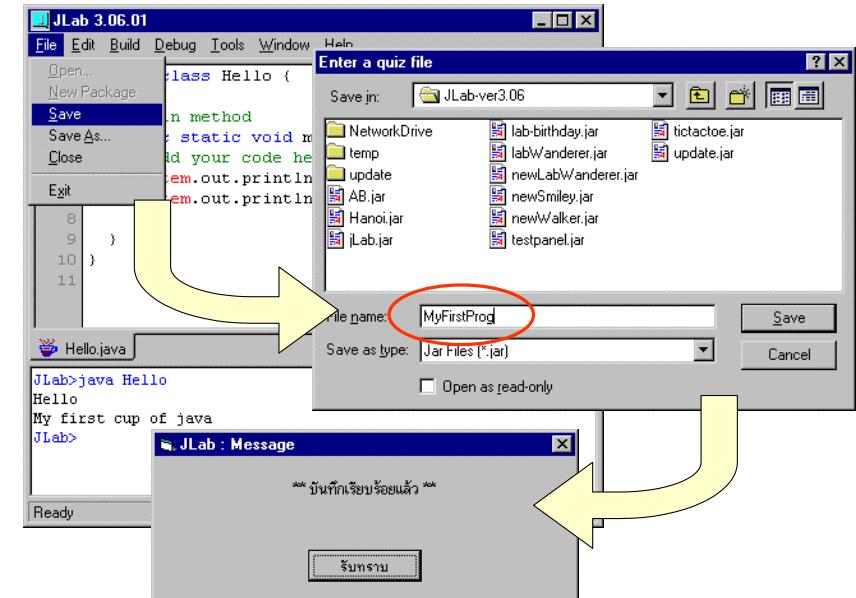
JLab : การสั่งโปรแกรมให้ทำงาน



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 29

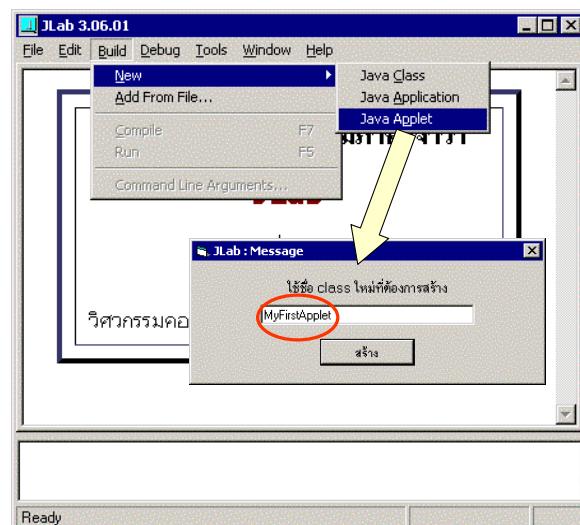
JLab : การบันทึก



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 30

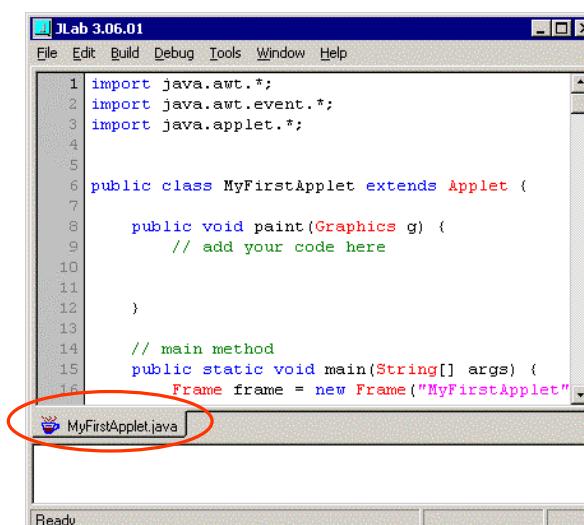
JLab : การสร้างโปรแกรมแบบกราฟิก



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 31

JLab : การสร้างโปรแกรมแบบกราฟิก



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 32

JLab : การเขียนโปรแกรม

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.applet.*;
4 
5 
6 public class MyFirstApplet extends Applet {
7 
8     public void paint(Graphics g) {
9         // add your code here
10        g.setColor(Color.red);
11        g.drawOval(0, 0, 100, 100);
12        g.fillOval(50, 50, 100, 100);
13    }
14 
15    // main method
16 }
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 33

JLab : การแปลงโปรแกรม

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.applet.*;
4 
5 
6 public class MyFirstApplet extends Applet {
7 
8     public void paint(Graphics g) {
9         // add your code here
10        g.setColor(Color.red);
11        g.drawOval(0, 0, 100, 100);
12        g.fillOval(50, 50, 100, 100);
13    }
14 
15    // main method
16 }
```

JLab>javac MyFirstApplet.java
JLab>

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 34

JLab : การสั่งโปรแกรมทำงาน

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.applet.*;
4 
5 
6 public class MyFirstApplet extends Applet {
7 
8     public void paint(Graphics g) {
9         // add your code here
10        g.setColor(Color.red);
11        g.drawOval(0, 0, 100, 100);
12        g.fillOval(50, 50, 100, 100);
13    }
14 
15    // main method
16 }
```

JLab>javac MyFirstApplet.java
JLab>

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 35

สรุป

- โปรแกรม Java ประกอบด้วย class
- การทำงานของ Java application จะเริ่มที่ main
- ควรเขียนโปรแกรมให้อ่านง่ายๆ
- บรรทัดคำสั่งต่างๆ ต้องปิดท้ายด้วย semicolon ;
- การทำงานจะทำทีละคำสั่งตามลำดับจากบนลงล่าง
- การทำโปรแกรมเป็นวงจรการ edit-compile-execute-debug

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 36

2110101 : การทำโปรแกรมคอมพิวเตอร์

ประเภทของข้อมูล ตัวแปร และตัวปฏิบัติการพื้นฐาน

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

การศึกษาภาษาการทำโปรแกรม

- ไวยากรณ์ของภาษา
 - เขียนอย่างไร ให้ถูกกฎ ตัวแปลภาษาเข้าใจ
- คำสั่ง
 - มีอะไรให้ใช้ ให้นำมาประกอบกัน (ตามไวยากรณ์)
เพื่อแก้ไขปัญหาที่ต้องการ
- ประเภทข้อมูล
 - เลือกใช้ประเภทข้อมูลให้เหมาะสมกับงาน
เพื่อประสิทธิภาพ และความถูกต้องในการทำงาน

หัวข้อ

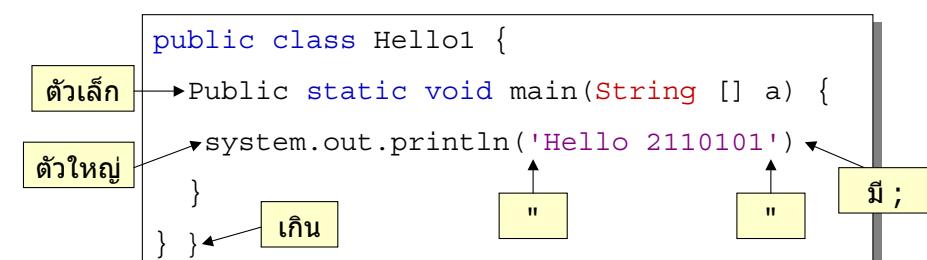
- สิ่งที่ต้องรู้ก่อนทำโปรแกรม
- ประเภทข้อมูล
 - จำนวนเต็ม
 - จำนวนจริง
- ตัวแปร
 - การประกาศ
 - กฎการตั้งชื่อ
- ตัวปฏิบัติการ + - * / %
- นิพจน์

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

13/5/02 2

ไวยากรณ์ของภาษาการทำโปรแกรม

- ไม่ซับซ้อน
- ไม่กำกวม
- แต่ผิดนิดผิดหน่อยไม่ได้ (ต้องเขียนถูก 100%)



compiler จะแจ้งตำแหน่งและสาเหตุของความผิดพลาด

คำสั่งและคลังคำสั่งของจาวา

- keywords : มีทั้งสิ้น 47 ตัว (เช่น do, while, if, else, switch, ...)
 - operators : มีวิธีปฏิบัติการทางคณิตศาสตร์และตรรกศาสตร์ทั้งสิ้น 40 วิธี (เช่น +, -, *, /, %, +=, ...)
 - library : มีคลังคำสั่งขนาดใหญ่ๆ ให้เรียกใช้ได้เป็นหมื่น (เช่น sqrt, cos, sort, clone, getTimeZone, getLocale, getWordInstance, ...)
- } 60%
} 1%

ประเภทข้อมูล

- ประเภทพื้นฐาน
 - มี 8 ประเภท
 - จำนวนเต็ม : `byte`, `short`, `int`, `long`
 - จำนวนจริง : `float`, `double`
 - อักขระ : `char`
 - ตรรกะ : `boolean`
 - ออกแบบเพิ่มเติมไม่ได้
- ประเภท class
 - มีแบบมาตรฐานเกือบสามพันแบบ
 - เช่น `String`, `Text`, `Rectangle`, ...
 - ออกแบบเองเพิ่มเติมได้

ประเภทข้อมูล : วัตถุประสงค์

- ระบบใช้เนื้อที่หน่วยความจำอย่างเหมาะสม
 - `byte` (1 ไบต์), `short` (2 ไบต์), `int` (4 ไบต์),
`long` (8 ไบต์), `float` (4 ไบต์), `double` (8 ไบต์),
`char` (2 ไบต์)
- ระบบประมวลผลอย่างมีประสิทธิภาพ
(เช่น ประมวลผลจำนวนเต็มแบบ `int` ไม่มากกว่าแบบ `long`)
- compiler ตรวจสอบความถูกต้องของโปรแกรมได้
- ผู้เขียนบอกจุดประสงค์ของการใช้ข้อมูล
- คนอื่นอ่านเข้าใจโปรแกรมได้ง่าย

ประเภทจำนวนเต็ม (integer)

- จำนวนเต็ม ("ไม่มีจุดทศนิยม")
เช่น 1023, 76289, 0, -10, +67392
- มีสีขนาด ขึ้นกับช่วงของจำนวนที่ใช้
 - `byte` (-128 ถึง 127)
 - `short` (-32768 ถึง 32767)
 - `int` (-2147483648 ถึง 2147483647)
 - `long` (- 2^{63} ถึง $2^{63} - 1$)

โดยทั่วไปใช้ `int`

ประเภทจำนวนจริง (real number)

- จำนวนที่มีจุดทศนิยมได้ เช่น 102.0, 3.14159, 0.0, -10.8, +0.3333
- จาวาเก็บจำนวนจริงแบบ floating point 10000000 เก็บ 0.1×10^8 , 0.0000012 เก็บ 0.12×10^{-5}
- มีสองขนาด ขึ้นกับความละเอียดของจำนวนที่เก็บ
 - `float` เก็บประมาณ 6-9 ตำแหน่งหลังจุดทศนิยมในช่วง (-3.4×10^{38} ถึง -1.4×10^{-45} ศูนย์ และ 1.4×10^{-45} ถึง 3.4×10^{38})
 - `double` เก็บประมาณ 15-17 ตำแหน่งหลังจุดทศนิยมในช่วง (-1.8×10^{308} ถึง -4.9×10^{-324} ศูนย์ และ 4.9×10^{-324} ถึง 1.8×10^{308})

โดยทั่วไปใช้ `double`

ทำไมต้องมีจำนวนเต็ม จำนวนจริงน่าจะพอ ?

- `float` เก็บข้อมูลในช่วงที่กว้างกว่า `int` 许多
- `int` เก็บข้อมูลได้ทุกๆ ตัวที่เป็นไปได้ในช่วง
- `float` เก็บได้บางตัวเท่านั้น (เพราะเก็บจำนวนตัวเลขหลังจุดทศนิยมได้จำกัด)
- ตัวอย่าง
 - `int` เก็บค่า 2147483647 ได้
 - `float` เก็บค่า 2147483647 ไม่ได้ ถ้ายัดเยียดให้ จะเก็บเป็น 2.14748365×10^9

ประเภทข้อมูลพื้นฐาน ประมวลผลได้เร็วมาก เพราะมีขนาดจำกัด และชาร์ดแวร์รองรับการทำงานโดยตรง

ประเภทอักขระ (character)

- ข้อมูลแบบ `char` คือตัวอักขระ หนึ่ง ตัว
- อักขระ (character) คือ
 - ตัวอักษร (ในภาษาต่างๆ) เช่น 'a', 'b', 'A', 'B', 'Ω', '₪', 'ก', 'ໝ'
 - ตัวเลข 0-9 (ในภาษาต่างๆ) เช่น '1', '2', '๑', '๒'
 - สัญลักษณ์พิเศษ (ในภาษาต่างๆ) เช่น '#', '\$', '฿', 'c~'
 - รหัสควบคุมอุปกรณ์ เช่น '\n' ตัวขึ้นบรรทัดใหม่ '\t' ตัว tab

ประเภทสายอักขระ (`String`)

- `String` คือลำดับของ `char` จำนวนศูนย์ตัวขึ้นไป
 - "engineering"
 - "webboy@universe.org"
 - "วิศวกรรมศาสตร์"
 - "Hello How are you?"
 - ""

"1004" ไม่เหมือนกับ 1004

"" มีความหมาย แต่ .. ไม่มี

ประเภทตรรกะ (boolean)

- boolean
- มีได้แค่สองค่าคือ true และ false เท่านั้น
- 1 ไม่ได้แทน true 0 ไม่ได้แทน false

ตัวแปร (Variable)

- ตัวแปรเป็นที่เก็บข้อมูลช่วงคราวของโปรแกรม
- ตัวแปรทุกตัวมีชื่อและประเภทข้อมูลกำกับ
- ต้องประกาศตัวแปรก่อนที่จะนำมาใช้

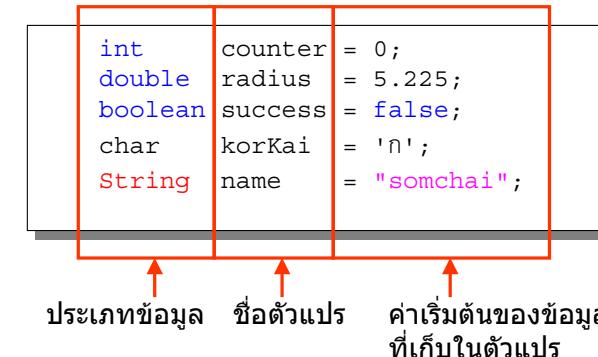
```
public class Zombie {  
  
    public static void main(final String[] args) {  
        int counter = 0;  
        double radius = 5.225;  
        boolean success = false;  
        String name = "somchai";  
  
        ...  
    }  
}
```

ตัวแปร

- ตัวแปรเป็นที่เก็บข้อมูลช่วงคราว
- การนำข้อมูลที่เก็บในตัวแปรมาใช้ ไม่ทำให้ข้อมูลที่เก็บในตัวแปรสูญหาย
- การนำข้อมูลใหม่เก็บใส่ตัวแปร เป็นการแทนที่ข้อมูลเดิม (ระบบไม่เก็บสำเนาข้อมูลเดิมไว้)

```
int numCircles = 27, k;           int numCircles   ??  
                                    int k  
  
System.out.println( numCircles );  
System.out.println( numCircles );  
  
k = 35;  
numCircles = k;                   35   35  
                                    int numCircles  int k
```

การประกาศตัวแปร



ประกาศบรรทัดละหลายตัว หรือจะไม่ใส่ค่าเริ่มต้นก็ได้

```
int i = 100, j = 200;  
double x0, y0, z0;  
double x1, y1 = 4.5, z1 = 7.8;
```

ตัวอย่างการประกาศตัวแปรที่ผิด

ไม่ได้ปิดท้ายบรรทัดด้วย ;

```
int counter
```

ใส่ค่าเริ่มต้นที่ผิดประเภท

```
int counter = 0.75;
double radius = "15.25";
boolean success = 1;
```

นิยามประเภทข้อมูลที่ไม่รู้จัก

```
integer counter;
doubleE radius = "15.25";
string title = "WWW";
```

ตั้งชื่อตัวแปรผิดกฎ

```
int boolean;
double public = 12.7;
String 12X = "WWW";
```

กฎการตั้งชื่อ

- ชื่อประกอบด้วยตัวอักษร ตัวเลข ตัว \$ หรือ _ ก็ได้
- ชื่อห้ามขึ้นต้นด้วยตัวเลข
- ชื่อยาวๆ ได้ไม่เป็นไร
- ตัวอักษรตัวใหญ่ไม่เหมือนตัวเล็ก
- ต้องไม่ซ้ำกับคำส่วนของภาษาจาวา
- ต้องไม่ซ้ำกับชื่ออื่นๆ ของโปรแกรม (เช่น ชื่อ class)

ตัวอย่างถูก

```
int17 butterCup Public int2String day_of_week
```

ตัวอย่างผิด

```
7zean I.love.you public ohOH! ed-edd-n-eddy
```

คำส่วนในภาษาจาวา

```
abstract assert boolean break byte
case catch char class const
continue default do double else
extends false final finally float
for goto if implements import
instanceof int interface long native
new null package private protected
public return short static strictfp
super switch synchronized this throw
throws transient true try void
volatile while
```

ห้ามนำคำส่วนมาตั้งเป็นชื่อตัวแปร

ธรรมเนียมนิยมในการตั้งชื่อ

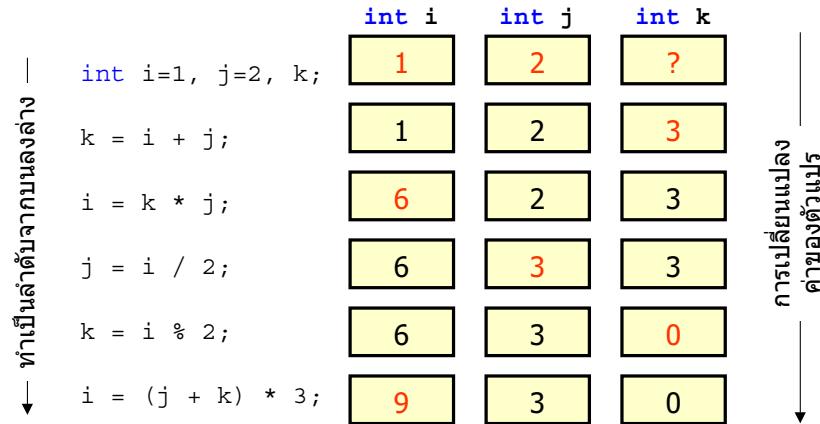
- ตัวแปร : ขึ้นต้นด้วยตัวอักษรตัวเล็ก
- class : ขึ้นต้นด้วยตัวอักษรตัวใหญ่
- หลีกเลี่ยงการใช้ _ และ \$
- สื่อความหมาย
- ถ้าชื่อประกอบด้วยคำหลายคำ เขียนแบบ "หลังอูฐ"

```
numStudents dayOfWeek imageBuffer
```

```
tvChannel customerID initialVelocity
```

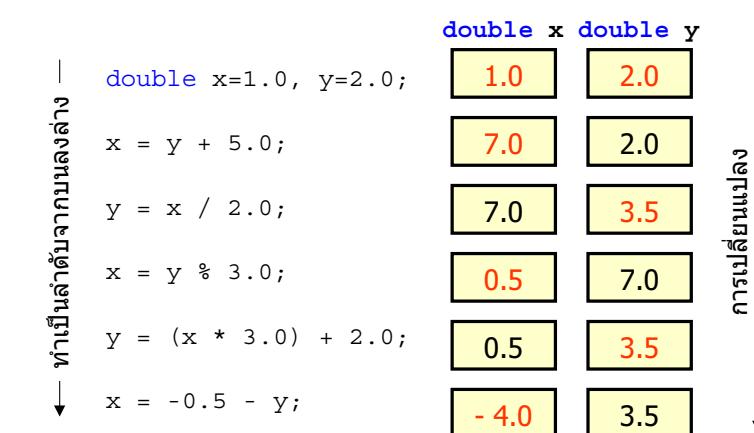
ตัวปฏิบัติการพื้นฐาน : จำนวนเต็ม

- บวก ลบ คูณ หาร และเศษเหลือจากการหาร
(+ - * / %)



ตัวปฏิบัติการพื้นฐาน : จำนวนจริง

- บวก ลบ คูณ หาร และเศษเหลือจากการหาร
(+ - * / %)



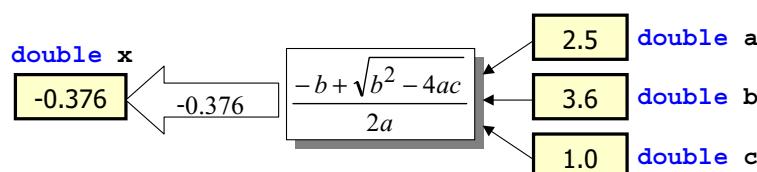
การให้ค่าตัวแปร (Assignment)

- ใช้เครื่องหมาย =

ตัวแปร = นิพจน์ ;

↑
ชื่อตัวแปรที่รับค่า
↑
นิพจน์ ที่ให้เป็นผลลัพธ์ซึ่งมีประเภท
ข้อมูลเดียวกับตัวแปรที่รับค่า

```
x = (-b + Math.sqrt(b*b - 4*a*c)) / (2.0*a);
```



นิพจน์ (Expression)

- ค่าคงตัว ตัวแปร พิงก์ชัน (หรือ method) หรือ
- การรวมกันของค่าคงตัว ตัวแปร พิงก์ชัน นิพจน์ย่ออยู่ด้วยตัวปฏิบัติการ และเครื่องหมายวงเล็บปิดเปิด ที่สามารถคำนวณค่าได้

$\frac{9c}{5} + 32$	$((9.0 * c) / 5.0) + 32.0$
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	$((9.0) * (c)) / (5.0) + (32.0)$
$9.0*c/5.0+32.0$	$-b + \sqrt{b^2 - 4ac}$
$(-b + Math.sqrt(b*b - 4.0*a*c)) / (2.0*a)$	$(-b + Math.sqrt(b*b - 4.0*a*c)) / (2.0*a)$

โดยที่ร้าวไปจะเดินข้องว่างเพื่อให้อ่านง่าย และเดินวงเล็บเพื่อกำจัดความก้าว

ตัวอย่าง

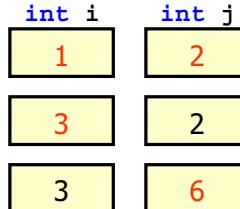
```
i = j;
```

นำค่าของตัวแปร *j* ไปใส่ในตัวแปร *i*

```
i = i + j;
```

นำค่าของตัวแปร *i* และ *j* มาบวกกัน
แล้วนำผลบวกที่ได้ไปใส่ในตัวแปร *i*

```
int i=1, j=2;
```



```
i = i + j;
```

```
j = j + j + j;
```

ตัวอย่าง

```
x ← -x
```

$$f \leftarrow \frac{9c}{5} + 32$$

$$c \leftarrow \frac{5(f-32)}{9}$$

$$d \leftarrow \sqrt{\Delta x^2 + \Delta y^2}$$

```
x = -x;
```

```
f = ((9.0 * c) / 5.0) + 32.0;
```

```
c = 5 * (f - 32.0) / 9.0;
```

```
d = Math.sqrt(dx*dx + dy*dy);
```

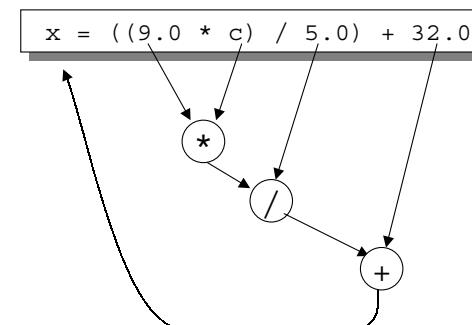
~~$$d \leftarrow \sqrt{\Delta x(\Delta x + \Delta y)\Delta y}$$~~

กฎการพิจารณาลำดับการคำนวณ

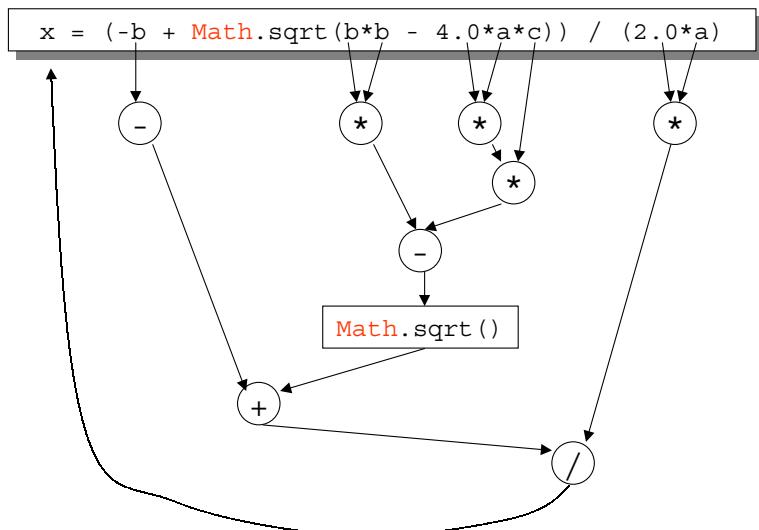
- ทำภายในวงเล็บหาก่อน
- ในกรณีมีตัวปฏิบัติการหลายตัว ตามลำดับดังนี้
 - การเรียกใช้ method
 - unary - +
 - * / %
 - + -
 - =
- จากซ้ายไปขวา (ในกรณีที่ operator มีสักติ์ครีเท่ากัน)

วงเล็บกำหนดลำดับการคำนวณ

```
x = ((9.0 * c) / 5.0) + 32.0
```



ตัวปฏิบัติการกำหนดลำดับการคำนวณ



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

13/5/02 29

ตัวอย่างผิดๆ

```
int i, k = 9;  
double x = 1.0;  
  
k = i;           // i is uninitialized  
i = x;          // assign double to int  
i = 3k;         // no operator  
3.0 * x + 5.78; // not a complete statement  
i = 3--k;       // should be i = 3 - (-k)  
i = (k + 5;     // incomplete parentheses
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

13/5/02 30

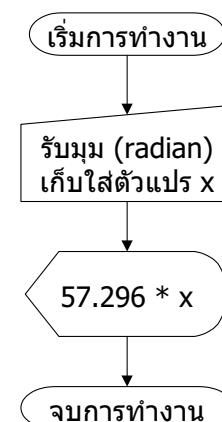
โปรแกรมแปลงหน่วยเรเดียนเป็นองศา

- Requirement
 - โปรแกรมรับมุมหน่วยเป็นเรเดียนแล้วแปลงเป็นองศา
- Analysis
 - รับมุม x เป็นจำนวนจริงผ่านทางแป้นพิมพ์
 - x เรเดียนมีค่าเท่ากับ $\frac{360x}{2\pi} \approx 57.296x$ องศา
 - แสดงมุมหน่วยเป็นองศาที่หาได้ทางจอภาพ
- Design
 - เขียนผังงาน
- Implementation
 - เขียนโปรแกรม

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

13/5/02 31

ผังงาน โปรแกรม และการทดสอบ



```
import jlab.JLabIO;  
  
public class Radian2Degree {  
  
    public static void main(String[] args) {  
        double x, d;  
        x = JLabIO.readDouble("input radian : ");  
        d = 57.296 * x;  
        System.out.println(d);  
    }  
}  
  
JLab>java Radian2Degree  
input radian : 3  
171.888  
JLab>
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

13/5/02 32

โปรแกรมแปลงหน่วยเรเดียนเป็นองศา

- ในJAVA มีค่า π ให้ใช้
- Math.PI = 3.14159265358979323846**

```
import jlab.JLabIO;

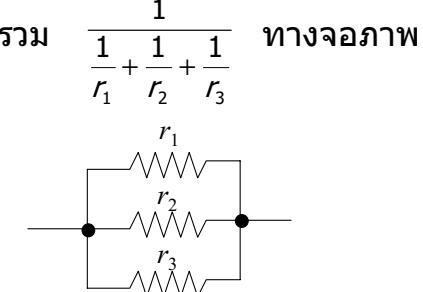
public class Radian2Degree {

    public static void main(String[] args) {
        double x, d;
        x = JLabIO.readDouble("input radian : ");
        d = 180.0 * x / Math.PI;
        System.out.println(x + " radians = " + d + " degrees");
    }
}
```

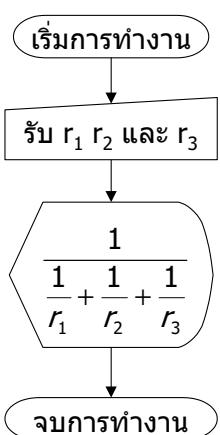
```
JLab>java Radian2Degree
input radian : 3
3.0 radians = 171.88733853924697 degrees
JLab>
```

โปรแกรมหา R จากการต่อ R 3 ตัวแบบขนาน

- Requirement**
 - โปรแกรมหาความต้านทานจากการต่อความต้านทาน 3 ตัวแบบขนาน
- Analysis**
 - รับค่าความต้านทาน ($K\text{ ohms}$) 3 ตัวจากแป้นพิมพ์
 - แสดงค่าความต้านทานรวม $\frac{1}{r_1} + \frac{1}{r_2} + \frac{1}{r_3}$ ทางจอภาพ
- Design**
 - เขียนผังงาน
- Implementation**
 - เขียนโปรแกรม



ผังงาน โปรแกรม และการทดสอบ



```
import jlab.JLabIO;

public class ParallelResistance {

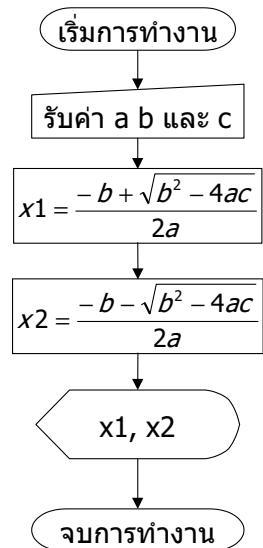
    public static void main(String[] args) {
        double r1, r2, r3, r;
        r1 = JLabIO.readDouble("r1 (Kohms) = ");
        r2 = JLabIO.readDouble("r2 (Kohms) = ");
        r3 = JLabIO.readDouble("r3 (Kohms) = ");
        r = 1.0 / (1.0/r1 + 1.0/r2 + 1.0/r3);
        System.out.println("R (parallel) = "
                           + r + " Kohms");
    }
}
```

```
JLab>java ParallelResistance
r1 (Kohms) = 22
r2 (Kohms) = 33
r3 (Kohms) = 100
R (parallel) = 11.660777385159012 Kohms
JLab>
```

โปรแกรมหารากของสมการกำลังสอง

- Requirement**
 - โปรแกรมหารากของสมการ $ax^2 + bx + c = 0$
- Analysis**
 - รับจำนวนจริง a b และ c ผ่านทางแป้นพิมพ์
 - คำนวณรากสองรากจากสูตร $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
 - แสดงรากทั้งสองที่หาได้ทางจอภาพ
- Design**
 - เขียนผังงาน
- Implementation**
 - เขียนโปรแกรม

ผังงาน



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

13/5/02 37

โปรแกรมและการทดสอบ

```
import jlab.JLabIO;  
  
public class QRoot {  
    public static void main(String[] args) {  
        double a, b, c, x1, x2;  
  
        a = JLabIO.readDouble("a = ");  
        b = JLabIO.readDouble("b = ");  
        c = JLabIO.readDouble("c = ");  
  
        double t = Math.sqrt(b * b - 4.0 * a * c);  
        x1 = (-b + t) / (2.0 * a);  
        x2 = (-b - t) / 2.0 / a;  
        System.out.println("x1 = " + x1);  
        System.out.println("x2 = " + x2);  
    }  
}  
  
x = 
$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
  
  
JLab>java QRoot  
a = 1  
b = 4  
c = 3  
x1 = -1.0  
x2 = -3.0  
JLab>
```

The code shows a Java program named QRoot that solves a quadratic equation. It uses the quadratic formula to calculate the roots x1 and x2 based on user input for coefficients a, b, and c. The program then prints the results. A mathematical formula for the quadratic root is also shown on the right side of the screen.

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

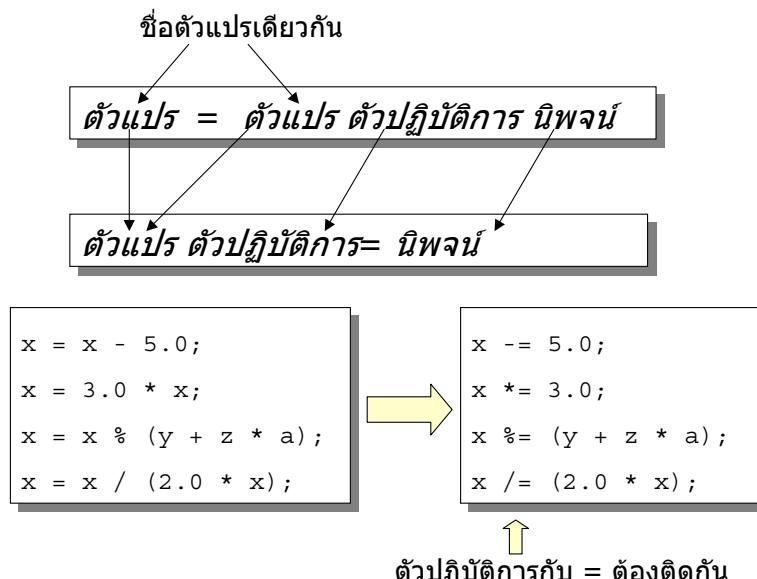
13/5/02 38

2110101 : การทำโปรแกรมคอมพิวเตอร์

ตัวปฏิบัติการพื้นฐาน การเปลี่ยนประเภทข้อมูล

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

วิธีเขียนลัด



หัวข้อ

- ตัวปฏิบัติการ =
- ตัวปฏิบัติการเพิ่ม และลดค่าข้อมูลอีกหนึ่ง
- การเขียนค่าคงตัวของจำนวนเต็มและจำนวนจริง
- การเปลี่ยนประเภทข้อมูล

โปรแกรมหาค่าเฉลี่ย

```
import jlab.JLabIO;  
  
public class Mean {  
    public static void main(String [] args) {  
        double x = 0.0;  
        x += JLabIO.readDouble("data #1 : ");  
        x += JLabIO.readDouble("data #2 : ");  
        x += JLabIO.readDouble("data #3 : ");  
        x += JLabIO.readDouble("data #4 : ");  
        x += JLabIO.readDouble("data #5 : ");  
        System.out.println("mean = " + (x / 5.0) );  
    }  
}
```

```
JLab>java Mean  
data #1 : 12  
data #2 : 35.5  
data #3 : 16.8  
data #4 : 19.0  
data #5 : 2.2  
mean = 17.1  
JLab>
```

ตัวปฏิบัติการพื้นฐาน : จำนวนเต็มและจริง (ต่อ)

• การเพิ่มค่าทีละหนึ่ง มีสองแบบ

- `a++` นำค่าของ `a` ไปใช้แล้วค่อยเพิ่มค่าของ `a` ขึ้นอีกหนึ่ง
- `++a` เพิ่มค่าของ `a` ขึ้นอีกหนึ่ง แล้วค่อยย้ายค่าไปใช้

int i	int j	int k
1	2	?
2	2	?
2	3	?
3	2	?
4	3	6
5	4	8

ทำเป็นลำดับจगวนลงล่าง ตามลำดับ
↑
| int i=1, j=2, k;
| i++;
| ++j;
| j = i++;
| k = ++j + i++;
↓ k = j++ + ++i;

การเปลี่ยนแปลงตัวแปร

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 5

ตัวปฏิบัติการพื้นฐาน : จำนวนเต็มและจริง (ต่อ)

• การลดค่าทีละหนึ่ง มีสองแบบ

- `a--` นำค่าของ `a` ไปใช้แล้วค่อยลดค่าของ `a` ลงหนึ่ง
- `--a` ลดค่าของ `a` ลงหนึ่ง แล้วค่อยย้ายค่าไปใช้

int i	int j	int k
10	4	?
9	4	?
9	3	?
8	8	?
7	7	0
6	6	1

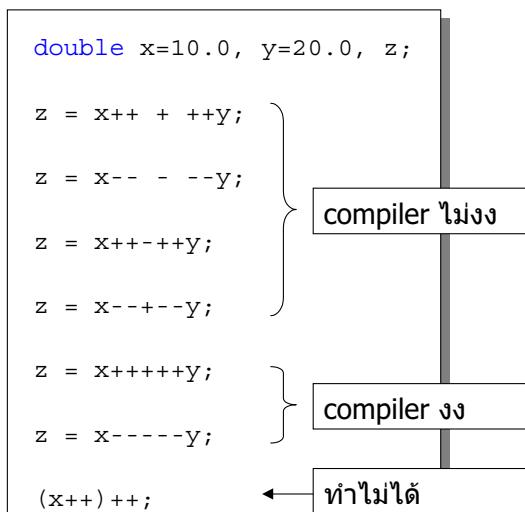
ทำเป็นลำดับจगวนลงล่าง ตามลำดับ
↑
| int i=10, j=4, k;
| i--;
| --j;
| j = --i;
| k = --j - --i;
↓ k = j--- --i;

การเปลี่ยนแปลงตัวแปร

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 6

ตัวอย่างงงงง



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 7

การเขียนค่าคงตัวที่เป็นจำนวนเต็ม

- ต้องไม่มี comma
- long ต้องปิดท้ายด้วย l หรือ L (แอลเล็กหรือใหญ่)
- ฐานสิบ : ห้ามเขียนเริ่มด้วย 0 (ยกเว้นค่า 0)
- ฐานแปด : ต้องเริ่มด้วย 0
- ฐานสิบหก : ต้องเริ่มด้วย 0x

```
int base10 = 2545;
int base8 = 04761;
int base16 = 0xF1;
```

```
long base10 = 2545L;
long base8 = 04761L;
long base16 = 0x9F1L;
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 8

การเขียนค่าคงตัวที่เป็นจำนวนจริง

- float ต้องปิดท้ายด้วย f หรือ F
- double ปิดท้ายด้วย d หรือ D หรือไม่มีก็ได้
- เขียนในรูปแบบย่อ ที่มีการคูณด้วย 10 ยกกำลังได้ เช่น
 - 1230000000.0 เขียนเป็น
 - 123E7 (ชึ้นแทน 123×10^7)
 - 12.3E8 (ชึ้นแทน 12.3×10^8)
 - 0.0123E11 (ชึ้นแทน 1.23×10^{11})
 - ...
 - -0.000567 เขียนเป็น
 - -567E-6 (ชึ้นแทน -567×10^{-6})
 - -5.67E-4 (ชึ้นแทน -5.67×10^{-4})
 - -0.567E-3 (ชึ้นแทน -0.567×10^{-3})
 - ...

ใช้ตัว e เล็กก์ได้

เลขชี้กำลังต้อง
เป็นจำนวนเต็ม

การเปลี่ยนประเภทของข้อมูล

- บางครั้งเราอาจต้องการเปลี่ยนข้อมูลประเภทหนึ่งไปยังอีกประเภทหนึ่ง เช่น
 - นิพจน์ที่ใช้มีค่าคงตัวและตัวแปรหลากหลายประเภท
 - ใช้ method ของระบบที่รับข้อมูลคล้ายประเภทกับที่เรามี
 - มีการนำจำนวนเต็มมาหารกัน แต่ต้องการผลซึ่งมีเศษด้วย
 - ผลที่ได้จากการคำนวณมีเศษ แต่ต้องการปัดทิ้ง
 - ...

```
int dx, dy, halfDist;  
...  
halfDist = 0.5 * Math.sqrt(dx*dx + dy*dy);
```

$$d = \left\lfloor \frac{\sqrt{\Delta x^2 + \Delta y^2}}{2} \right\rfloor$$

Type Checking

- การให้ค่าผิดประเภทกับตัวแปรอาจแปลไม่ผ่าน

```
public class TypeChecking {  
    public static void main(String[] args) {  
        double x = 1.0;  
        short k = 100000;  
        float f = 1.2;  
    }  
}
```

JLab>javac TypeChecking
TypeChecking.java:4: [#1102] [#1155] อาจสูญเสียความ
แม่นยำของข้อมูลได้ พน int แต่ต้องการ short, column:15
TypeChecking.java:5: [#1102] [#1155] อาจสูญเสียความ
แม่นยำของข้อมูลได้ พน double แต่ต้องการ float, column:15
JLab>

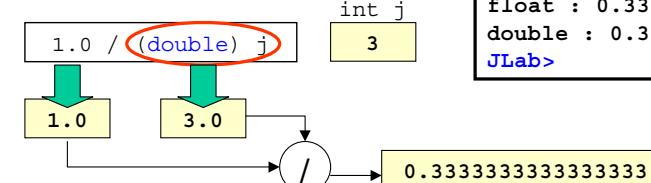
compilation error

การเปลี่ยนประเภทของข้อมูล

- cast operator : ใส่ชื่อประเภทใหม่ภายใต้วงเล็บไว้หน้านิพจน์ที่ต้องการให้เปลี่ยน

```
public class TypeCasting {  
    public static void main(String[] args) {  
        int j = 3;  
        System.out.println("int : " + (1 / j));  
        System.out.println("float : " + (1.0F / (float) j));  
        System.out.println("double : " + (1.0 / (double) j));  
    }  
}
```

JLab>java TypeCasting
int : 0
float : 0.33333334
double : 0.3333333333333333
JLab>

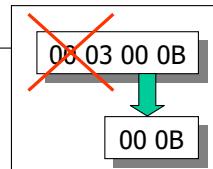


ข้อมูลอาจสูญหายในการเปลี่ยนประเภทข้อมูล

- เปลี่ยนจำนวนจริงเป็นจำนวนเต็ม เช่นจะถูกตัดทิ้ง
- เปลี่ยนไปเป็นประเภทที่รับช่วงข้อมูลได้แคบกว่า

```
public class LossFromTypeCasting {  
    public static void main(String[] args) {  
        int i = (int) 12.999;  
        int k = 196620; // 0x3000B  
        short tiny = (short) k;  
        System.out.println("i = " + i);  
        System.out.println("tiny = " + tiny);  
    }  
}
```

narrowing primitive conversion



```
JLab>java LossFromTypeCasting  
i = 12  
tiny = 11  
JLab>
```

ระวังเรื่องลำดับการทำงาน

- cast operator จะถูกทำก่อน + - * / %
แต่หลัง () และ unary + -

```
public class PrecedenceOfCasting {  
    public static void main(String[] args) {  
        double x = 1.9, y = 10.9;  
        System.out.println("(int) x * y = " + ((int) x * y));  
        System.out.println("(int) y * x = " + ((int) y * x));  
        System.out.println("(int) (x * y) = " + ((int) (x * y)));  
        System.out.println("(int) (y * x) = " + ((int) (y * x)));  
    }  
}
```

```
JLab>java PrecedenceOfCasting  
(int) x * y = 10.9  
(int) y * x = 19.0  
(int) (y * x) = 20  
(int) (x * y) = 20  
JLab>
```

Narrowing Primitive Conversion

- byte เป็น char
- short เป็น byte หรือ char
- char เป็น byte หรือ short
- int เป็น byte, short, หรือ char
- long เป็น byte, short, char, หรือ int
- float เป็น byte, short, char, int, หรือ long
- double เป็น byte, short, char, int, long, หรือ float

การเปลี่ยนประเภทของข้อมูลอย่างอัตโนมัติ

- เมื่อให้ค่ากับตัวแปรที่รับช่วงข้อมูลได้กว้างกว่า
- + - * / หรือ % ในนิพจน์รับข้อมูลต่างประเภทกัน
 - ถ้ามีข้อมูลหนึ่งเป็น double จะเปลี่ยนอีกด้วยเป็น double ด้วย
 - ไม่เช่นนั้น ถ้ามีข้อมูลหนึ่งเป็น float จะเปลี่ยนอีกด้วยเป็น float ด้วย
 - ไม่เช่นนั้น ถ้ามีข้อมูลหนึ่งเป็น long จะเปลี่ยนอีกด้วยเป็น long ด้วย
 - ไม่เช่นนั้น เปลี่ยนข้อมูลทั้งคู่เป็น int

```
public class AutoConversion {  
    public static void main(String[] args) {  
        int i = 7, j = 8;  
        double x = 0.5F * i + j;  
        System.out.println("x = " + x);  
    }  
}
```

```
JLab>java AutoConversion  
x = 11.5  
JLab>
```

widening primitive conversion

Widening Primitive Conversion

- byte เป็น short, int, long, float, หรือ double
- short เป็น int, long, float, หรือ double
- char เป็น int, long, float, หรือ double
- int เป็น long, float, หรือ double
- long เป็น float หรือ double
- float เป็น double

อาจทำให้หลักท้ายๆ ของจำนวนสูญหายได้ เมื่อ

- เปลี่ยน int หรือ long เป็น float
- เปลี่ยน long เป็น double

```
public class Widening {  
    public static void main(String[] args) {  
        int big = 1234567890;  
        float approx = big;  
        System.out.println("big = " + big);  
        System.out.println("approx = " + approx);  
    }  
}
```

```
JLab>java Widening  
big = 1234567890  
approx = 1.23456794E9  
JLab>
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 17

คำถาม

double x = (1/2 + 1/2);

x มีค่าเท่าใด ?

float y = 1.0/2.0;

y มีค่าเท่าใด ?

double z = 4d + 5;

z มีค่าเท่าใด ?

double u = 0.0 + 1/2;

u มีค่าเท่าใด ?

double v = (0.0 + 1)/2;

v มีค่าเท่าใด ?

double f = 212.0;

double c = (5/9) * (f - 32);

c มีค่าเท่าใด ?

เรื่องต้องรู้

floating point มีขนาดจำกัด

```
public class Imprecision1 {  
    public static void main(String[] args) {  
        double x = 20.1;  
        double y = x + 16.8;  
        System.out.println(y);  
    }  
}
```

```
JLab>java Imprecision1  
36.900000000000006  
JLab>
```

ลำดับการคำนวณนั้นสำคัญ

```
public class Imprecision2 {  
    public static void main(String[] args) {  
        double x = 1.0E-200;  
        System.out.println(x * x / x);  
        System.out.println(x * (x / x));  
    }  
}
```

```
JLab>java Imprecision2  
0.0  
1.0E-200  
JLab>
```

เรื่องต้องรู้

```
public class DivByZero {  
    public static void main(String[] args) {  
        int i = 0;  
        int j = 5 / i;  
  
        System.out.println(j);  
    }  
}
```

```
JLab>java DivByZero  
java.lang.ArithmetricException: / by zero  
at DivByZero.main(DivByZero.java:6)  
Exception in thread "main"  
JLab>
```

จำนวนเต็มในภาษาไม่มีค่า infinity
ดังนั้นการหารด้วยศูนย์จะทำให้เกิดความผิดพลาด
ขณะทำงาน (เรียกว่าเกิด exception)

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 19

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 20

เรื่องต้องรู้

```
public class DivByZero {  
    public static void main(String[] args) {  
        double i = 0.0;  
        double j = 5.0 / i;  
        System.out.println(j);  
        System.out.println(0.0/i);  
    }  
}
```

```
JLab>java DivByZero  
Infinity  
NaN  
JLab>
```

floating point ในจาวามีค่า infinity
และ NaN (Not a Number)
ดังนั้นการหารด้วยศูนย์จะไม่เกิด exception

2110101 : การทำโปรแกรมคอมพิวเตอร์

String และ I/O พื้นฐาน

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อ

- ข้อมูลประเภท **char**
- ข้อมูลประเภท **String**
 - methods ที่เกี่ยวกับ **String** และ + operator
- Console Input / Output
 - System.out.print**, **System.out.println**
 - คลังคำสั่ง JLab I/O

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 2

ข้อมูลประเภทอักขระ (**char**)

- ข้อมูลแบบ **char** คือตัวอักขระ **หนึ่ง** ตัว
- อักขระ (character) คือ
 - ตัวอักษร (ในภาษาต่างๆ)
เช่น 'a', 'b', 'A', 'B', 'Ω', 'ก', 'ຂ'
 - ตัวเลข 0-9 (ในภาษาต่างๆ)
เช่น '1', '2', '๑', '๒'
 - สัญลักษณ์พิเศษ (ในภาษาต่างๆ)
เช่น '#', '\$', '฿', 'ػ'
 - รหัสควบคุมอุปกรณ์
เช่น '\n' ตัวขึ้นบรรทัดใหม่ '\t' ตัว tab

การแทนอักขระด้วยรหัส

- จาwake กับอักขระต่างๆ ด้วยการใช้รหัสมาตรฐาน (ที่เรียกว่า Unicode) ขนาด 16 บิต อักขระละหนึ่งรหัส เช่น
 - 'a' (รหัส 0061_H), 'b' (รหัส 0062_H), 'A' (รหัส 0041_H), 'B' (รหัส 0042_H), 'Ω' (รหัส 03A9_H), 'ػ' (รหัส 0E58_H), 'ກ' (รหัส 0E01_H), 'ຂ' (รหัส 0E02_H)
- รองรับสารพัดภาษา
(internationalization เขียนย่อว่า i18n)

การเขียนค่าคงตัวแบบ `char`

- ค่าคงตัวแบบ `char` เขียนภายใต้เครื่องหมาย ''
- อักษรพิเศษบางตัว ต้องใช้ escape sequence (เริ่มด้วย \) เช่น
 - '\n' ตัวขึ้นบรรทัดใหม่
 - '\t' ตัว tab
 - '\\' สัญลักษณ์ \ (backslash)
 - '\'' สัญลักษณ์ ' (single quote)
 - '\"' สัญลักษณ์ " (double quote)

ข้อมูลประเภทสายอักษร (`String`)

- `String` เขียนเริ่มด้วย S ตัวใหญ่
- ข้อมูลแบบ `String` เก็บตัวอักษรได้หลายตัว
- ค่าคงตัวแบบ `String` เขียนภายใต้เครื่องหมาย "
- เขียนค่าคงตัวแบบ `String` ข้ามบรรทัดไม่ได้
- เรียกจำนวนอักษรว่า **ความยาว**
- เรียก "" ว่า null string

```
public class StringSample {  
    public static void main(String [] args) {  
        String s1 = "This is a string literal";  
        String s2 = "This is an illegal  
                    string literal";  
    }  
}
```

2 ไม่เหมือนกับ '2'

- จำนวนเต็ม 2 `short` 00 02
 - ใช้กับการประมวลผลเชิงคณิตศาสตร์ (+ - * / % ...)
- อักษร '2' `char` 00 32
 - ใช้สื่อสารระหว่างอุปกรณ์ แสดงทางจอภาพ หรือประมวลผลเชิงอักษรอื่นๆ
 - การประมวลผลเชิงคณิตศาสตร์กับอักษร จะไม่ได้ผลอย่างที่คิด ('2' + '3' ไม่ได้ '5')
 - ถ้าจะนำไปประมวลผลเชิงคณิตศาสตร์ก็ต้องเปลี่ยนเป็นจำนวนเต็มเสียก่อน

ข้อสังเกต : มีจำนวนเต็ม 2 แต่ไม่มี character '-2'

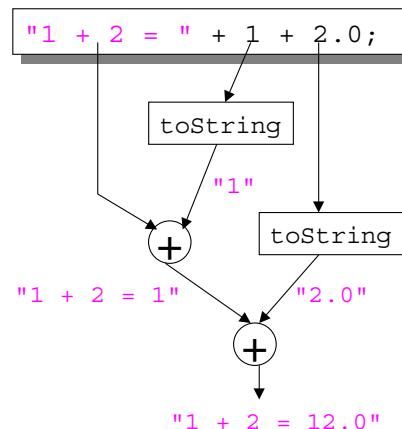
ตัวปฏิบัติการ + กับ `String`

- `String` มีตัวปฏิบัติการตัวเดียวคือ +
- + คือการต่อสายอักษรสองสายเข้าด้วยกัน
- เมื่อข้อมูลแบบ `String` + กับข้อมูลใด ข้อมูลนั้นจะถูกเปลี่ยนเป็น `String` ก่อน แล้วจึง +

```
public class StringConcatenation {  
    public static void main(String [] args) {  
        String s1 = "Approx. value of PI";  
        String s2 = "is";  
        System.out.println(s1 + s2);  
        System.out.println(s1 + " " + s2);  
        System.out.println(s1 + " " + s2 + " " + Math.PI);  
    }  
}
```

```
JLab>java StringConcatenation  
Approx. value of PIis  
Approx. value of PI is  
Approx. value of PI is 3.141592653589793  
JLab>
```

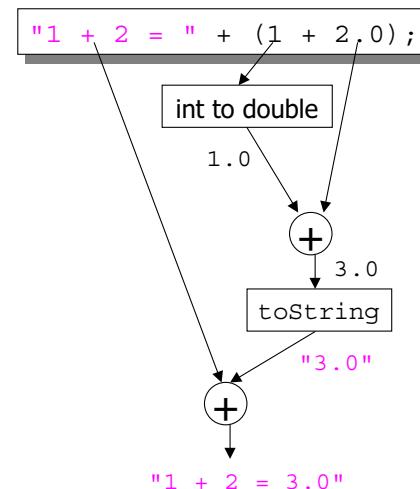
ข้อควรระวัง : ลำดับการ +



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 9

ข้อควรระวัง : ลำดับการ +



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 10

String ไม่ใช้ประเภทข้อมูลพื้นฐานใน Java

- **String** เป็น class มาตรฐานใน Java
- มี method ให้เรียกใช้มากมาย
- วิธีเรียกใช้ method ของ **String**
 - ชื่อตัวแปรที่เป็น String
 - ตามด้วยเครื่องหมายจุด
 - ตามด้วยชื่อ method
 - ตามด้วย()
 - ภายในวงเล็บต้องใส่อะไร ขึ้นกับ method ที่ใช้

```
String s1 = "Java JA";
String s2 = s1.toUpperCase(); // "JAVA JA"
char c = s2.charAt(2);      // 'V'
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 11

บาง methods ของ String

- length() คืน int ชี้สิ่งแทนความยาวของสายอักขระนี้
- toUpperCase() คืน String ซึ่งมีค่าเหมือนสายอักขระนี้ แต่ตัวอักษรอังกฤษทุกตัวเปลี่ยนเป็นตัวใหญ่
- trim() คือ String ซึ่งมีค่าเหมือนสายอักขระนี้ แต่ตัด spaces ที่อยู่ทั้งซ้ายและทางขวาออก

```
public class StringAPI {
    public static void main(String [] args) {
        String s1 = " Moo Mak ";
        System.out.println(s1.length());
        System.out.println(" " + s1.toUpperCase() + " ");
        System.out.println(" " + s1.trim() + " ");
    }
}
```

10
" MOO MAK "
"MOO MAK"

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 12

บาง methods ของ **String** (ต่อ)

- **substring(int b, int e)**
 - รับ **int** สองจำนวน (สมมติให้ชื่อเป็น b และ e)
 - คืน **String** ที่เป็นสายอักขระย่ออยเริ่มตั้งแต่ตัวที่ b เป็นจำนวน e - b ตัว
 - ตำแหน่งซ้ายสุดของสายอักขระคือตำแหน่ง 0

```
public class Substring {  
    public static void main(String [] args) {  
        String s1 = "Scorpian";  
        System.out.println(s1.substring(1,5));  
    }  
}  
  
0 1 2 3 4 5 6 7  
S c o r p i a n  
JLab>java Substring  
corp  
JLab>
```

Java API Help File

- ต้องหัดใช้และอ่าน help file ให้เป็น
- จำว่ามีบริการให้ใช้มากมาย จำไม่ได้หมด ต้องค้นและอ่านให้เป็น เมื่อต้องการใช้
- จะให้ฝึกในปฏิบัติการ

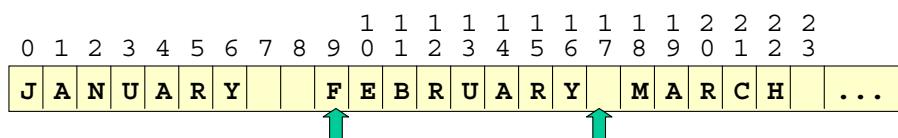
Methods ต่างๆ ของ **String**

- **String** มี methods มากมายให้ใช้
- อยากรู้รายละเอียดต้องอ่าน help file ของ Java API
- ใน JLab กดปุ่ม F1 แล้วให้ระบบหาคำว่า String

charAt	compareTo	concat
contentEquals	copyValueOf	endsWith
equals	getBytes	getChars
hashCode	indexOf	intern
lastIndexOf	length	matches
regionMatches	replace	replaceAll
replaceFirst	split	startsWith
subSequence	substring	toCharArray
toLowerCase	toString	toUpperCase
trim	valueOf	

โปรแกรมแสดงชื่อเดือน

- **Requirement**
 - แสดงชื่อเดือนจากหมายเลขเดือนที่ได้รับ
- **Analysis**
 - รับหมายเลขของเดือนผ่านทางแป้นพิมพ์
 - แสดงชื่อเดือนทางจอภาพ
 - ข้อสังเกต : เดือนที่ชื่อยาวสุดคือ "**SEPTEMBER**"
- **Design**
 - สร้างสายอักขระยาว $9 \times 12 = 108$ ตัว บรรจุชื่อเดือนเรียงต่อกันไป เดือนละ 9 ตัวอักษร



ผังงานการแสดงชื่อเดือน



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 17

คำสั่ง console I/O พื้นฐานใน Java

- แสดงออกจากการ
ใช้ **System.out.print()**, **System.out.println()**
- รับข้อมูลจากแป้นพิมพ์
ใช้ **System.in** แต่วิธีใช้จะยุ่งยากเล็กน้อย

```
// read a string from standard input
BufferedReader reader = new BufferedReader(
    new InputStreamReader(System.in));
String s = "";
try {
    s = reader.readLine();
} catch (IOException e) {
    e.printStackTrace();
}
```

คงยังอ่านไม่รู้เรื่อง
ไม่เป็นไร

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 19

โปรแกรมพิมพ์ชื่อเดือน

```
import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "";
        month += "JANUARY FEBRUARY MARCH APRIL ";
        month += "MAY JUNE JULY AUGUST ";
        month += "SEPTEMBER OCTOBER NOVEMBER DECEMBER ";
        int m = JLabIO.readInt("Enter month number : ");
        System.out.println(month.substring(9*(m-1), 9*m));
    }
}
```

JLab>java Month
Enter month number : 2
FEBRUARY
JLab>

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
J	A	N	U	A	R	Y		F	E	B	R	U	A	R	Y	M	A	R	C	H		...	

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 18

คำสั่ง console input ที่มีให้เฉพาะใน JLab

- JLabIO.readInt()** : รอรับจำนวนเต็ม
- JLabIO.readDouble()** : รอรับจำนวนจริง
- JLabIO.readString()** : รอรับสายอักขระ
- ภายในวงล้อจะใส่หรือไม่ใส่ **String** ก็ได้
ถ้าใส่ จะแสดงออกจากการก่อนรับข้อมูล
- จะต้องเติมบรรทัด **import jlab.JLabIO;**
ไว้ที่บรรทัดแรกๆ ของโปรแกรม

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 20

คำสั่ง console input ที่มีให้เฉพาะใน JLab

```
import jlab.JLabIO;
public class Month {
    public static void TestJLabIO(String [] args) {
        String name = JLabIO.readString("Name : ");
        int age = JLabIO.readInt("Age : ");
        double tax = JLabIO.readDouble("Income Tax : ");
        System.out.println(name + ", age = " + age
                           + ", tax = " + tax);
    }
}
```

JLab พองเงย →
JLab>java TestJLabIO
Name : Somchai
Age : 41
Income Tax : I won't tell you
invalid real number, try again
15000.50
Somchai, age = 41, tax = 15000.5
JLab>

Bar code : มาตรฐาน EAN-13

- เป็นมาตรฐาน bar code ประกอบด้วยตัวเลข 13 ตัว
- ตัวทางขวาสุดคือตัวเลขพิเศษ (check digit) ซึ่งถูกเพิ่มเพื่อใช้ตรวจสอบความถูกต้อง 12 ตัวทางซ้าย
- กำหนดให้ ตัวเลขทางขวาสุดเป็นเลขลำดับที่ 0

$$C = \left(10 - \left(\sum_{\substack{0 \leq k \leq 11 \\ k \text{ is even}}} d_k + 3 \times \sum_{\substack{0 \leq k \leq 11 \\ k \text{ is odd}}} d_k \right) \bmod 10 \right) \bmod 10$$

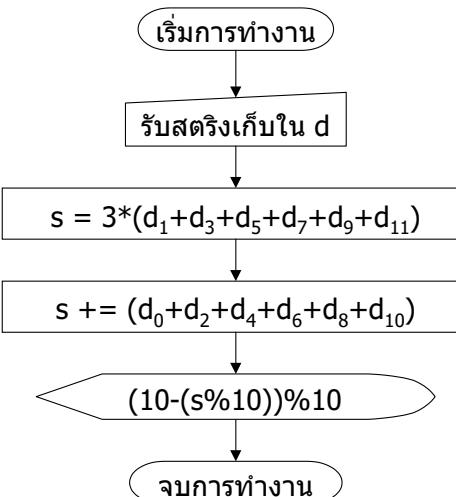


$$\begin{aligned} ((7+1+3+5+7+9)+3(0+2+4+6+8+0)) &= 92 \\ (10-(92\%10))\%10 &= 8 \end{aligned}$$

โปรแกรมหา check digit ของ EAN-13

- Requirement
 - แสดง check digit ของตัวเลข 12 หลักในระบบ EAN-13
- Analysis
 - รับตัวเลข 12 หลักทางแป้นพิมพ์
 - ใช้สูตรการหา check digit ของ EAN-13
 - แสดง check digit ทางจอภาพ
- Design
 - เขียนผังงาน
- Implementation
 - เขียนโปรแกรม

ผังงาน



โปรแกรมหา check digit ของ EAN-13

```
import jlab.JLabIO;
public class EAN13CheckDigit {
    public static void main(String[] args) {
        int s = 0;
        String d = JLabIO.readString("Enter 12 digits : ");
        s += Character.digit(d.charAt(1), 10);
        s += Character.digit(d.charAt(3), 10);
        s += Character.digit(d.charAt(5), 10);
        s += Character.digit(d.charAt(7), 10);
        s += Character.digit(d.charAt(9), 10);
        s += Character.digit(d.charAt(11), 10);
        s *= 3;
        s += Character.digit(d.charAt(0), 10);
        s += Character.digit(d.charAt(2), 10);
        s += Character.digit(d.charAt(4), 10);
        s += Character.digit(d.charAt(6), 10);
        s += Character.digit(d.charAt(8), 10);
        s += Character.digit(d.charAt(10), 10);
        System.out.println("Check Digit is " + (10-(s%10))%10);
    }
}
```

2110101 : การทำโปรแกรมคอมพิวเตอร์

Selection : if, if-else, switch-case

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อ

- คำสั่ง if
- คำสั่ง if-else
- Boolean expression
 - Relational operator
 - Boolean operator
 - operator precedence
- Nested if
- Statement & block
- switch-case
- การเปรียบเทียบจำนวนจริงและสตริง

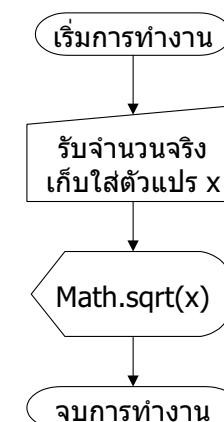
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 2

โปรแกรมหารากที่สอง

- Requirement
 - อ่านจำนวนจากผู้ใช้ และแสดงรากที่สองของจำนวนนั้น
- Analysis
 - รับจำนวนจริงผ่านทางแป้นพิมพ์
 - ใช้ **Math.sqrt** ในการหารากที่สอง
 - แสดงรากที่สองที่หาได้ทางจอภาพ
- Design
 - เขียนผังงาน
- Implementation
 - เขียนโปรแกรม

ผังงาน โปรแกรม และการทดสอบ



```
import jlab.JLabIO;
public class Sqrt {
    public static void main(String [] args) {
        double x = JLabIO.readDouble("x = ");
        System.out.println("sqrt(x) = " +
                           Math.sqrt(x));
    }
}
JLab>java Sqrt
x = 1.44
sqrt(x) = 1.2
JLab>
```

การทดสอบ

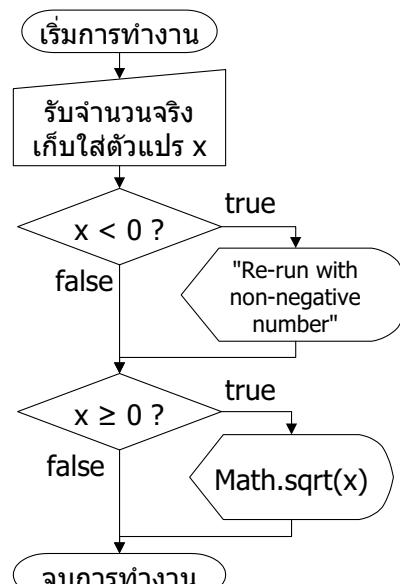
- จะเกิดขึ้นถ้าผู้ใช้ป้อนจำนวนติดลบ

```
import jlab.JLabIO;  
  
public class Sqrt {  
    public static void main(String [] args) {  
        double x = JLabIO.readDouble("x = ");  
        System.out.println("sqrt(x) = " +  
                           Math.sqrt(x));  
    }  
}
```

JLab>java Sqrt
x = -8
sqrt(x) = NaN
JLab>

เมื่อ $x < 0$ `Math.sqrt(x)` จะคืนค่าที่มีความหมายว่าไม่ใช่จำนวน (not a number) ค่านี้จะถูกเปลี่ยนเป็น "NaN" เมื่อแสดงทางจอภาพ ผู้ใช้ที่ไม่เข้าใจความหมาย NaN ควรแก้โปรแกรมให้สื่อความหมาย

ผังงาน



โปรแกรมหารากที่สอง (ฉบับปรับปรุง)

• Requirement

- อ่านจำนวนจากผู้ใช้ และแสดงรากที่สองของจำนวนนั้น ถ้าเป็นจำนวนไม่ติดลบ

• Analysis

- รับจำนวนจริงผ่านทางแป้นพิมพ์
- ถ้าติดลบ ให้แสดง "Re-run with non-negative number"
- ถ้าเป็นบวกหรือ 0 ให้แสดงรากที่สองที่หาได้ทางจอภาพ

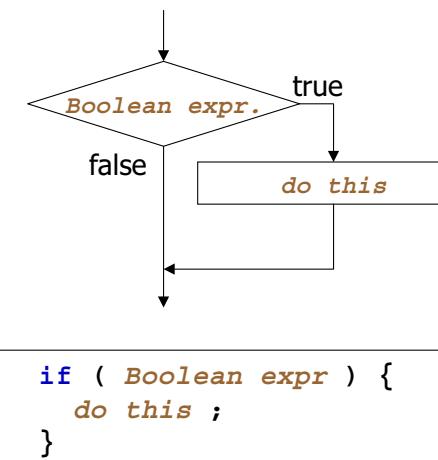
• Design

- เขียนผังงาน

• Implementation

- เขียนโปรแกรม

Selection : if statement



โปรแกรม และการทดสอบ

```
import jlab.JLabIO;

public class Sqrt {
    public static void main(String [] args) {
        double x = JLabIO.readDouble("x = ");
        if (x < 0) {
            System.out.println("Re-run with non-negative number");
        } else {
            System.out.println("sqrt(x) = " + Math.sqrt(x));
        }
    }
}
```

JLab>java Sqrt
x = -8
Re-run with non-negative number
JLab>

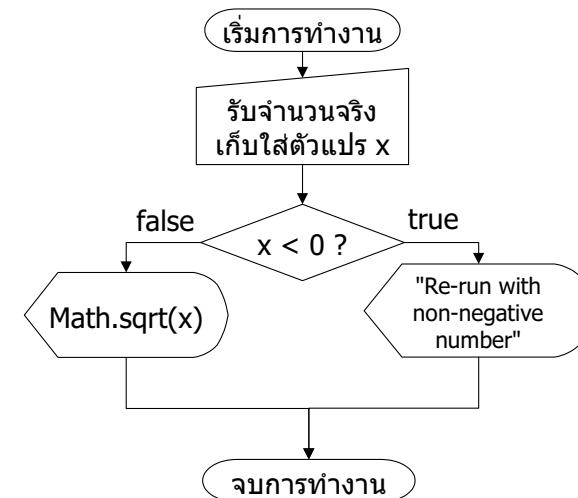
JLab>java Sqrt
x = 1.21
sqrt(x) = 1.1
JLab>

ข้อสังเกต : สองกรณีนี้ต้องเป็นจริงกรณีได้กรณีหนึ่งแน่ๆ (และไม่มีทางเป็นจริงพร้อมกันทั้งคู่)

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 9

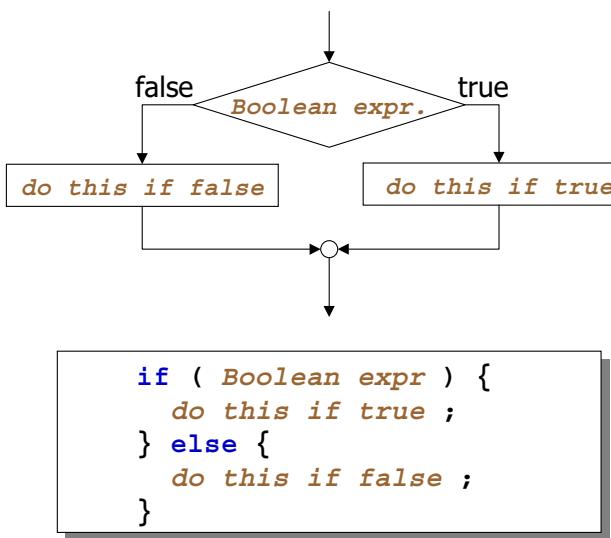
ผังงาน



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 10

Selection : if-else statement



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 11

โปรแกรม และการทดสอบ

```
import jlab.JLabIO;

public class Sqrt {
    public static void main(String [] args) {
        double x = JLabIO.readDouble("x = ");
        if (x < 0) {
            System.out.println("Re-run with non-negative number");
        } else {
            System.out.println("sqrt(x) = " + Math.sqrt(x));
        }
    }
}
```

JLab>java Sqrt
x = -8
Re-run with non-negative number
JLab>

JLab>java Sqrt
x = 1.21
sqrt(x) = 1.1
JLab>

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 12

นิพจน์บูลีน (Boolean Expression)

- ได้ผลลัพธ์เป็นประเภท boolean
- สร้างมาจากการใช้
 - relational operator : ตัวเปรียบเทียบข้อมูลแบบพื้นฐาน เช่นมากกว่า น้อยกว่า เท่ากัน
 - method ที่คืนค่าประเภท boolean
 - boolean operator : ตัวเชื่อมความสัมพันธ์นิพจน์บูลีน เช่น and or not
- boolean เปลี่ยนเป็น String ได้เท่านั้น

```
public class TrueFalse {  
    public static void main(String [] args) {  
        System.out.println( true + " " + false );  
    }  
}
```

JLab>java TrueFalse
true false
JLab>

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 13

Boolean Operators

- ใช้เครื่องหมาย && แทน and
- ใช้เครื่องหมาย || แทน or
- ใช้เครื่องหมาย ! แทน not
- and or เป็นตัวปฏิบัติการเพื่อเชื่อมนิพจน์บูลีนสองนิพจน์ ได้เป็นนิพจน์บูลีนใหม่
 - ให้ A และ B เป็นนิพจน์บูลีน
 - A && B เป็น true เมื่อ ทั้ง A และ B เป็น true ทั้งคู่
 - A || B เป็น true เมื่อ A หรือ B เป็น true ลักษณะดัง
- not เป็นตัวปฏิบัติการเพื่อเปลี่ยนผลของนิพจน์บูลีนที่กำหนดให้ จากจริงเป็นเท็จ หรือจากเท็จเป็นจริง

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 15

Relational Operators

- ตัวเปรียบเทียบข้อมูลประเภทพื้นฐาน ได้แก่
 - < (น้อยกว่า) > (มากกว่า)
 - <= (น้อยกว่าหรือเท่ากับ) >= (มากกว่าหรือเท่ากับ)
 - == (เท่ากัน) != (ไม่เท่ากัน)
- ข้อมูลที่นำมาเปรียบเทียบต้องเป็นประเภทเดียวกัน หรือใช้หลักการและวิธีการเปลี่ยนประเภทข้อมูล เช่น เดียวกับนิพจน์ทางคณิตศาสตร์
- ได้ผลลัพธ์เป็นประเภท boolean

((n % 2) == 1) ได้ค่าเป็น

- true ถ้า n เป็นจำนวนคี่
- false ถ้า n เป็นจำนวนคู่

n = 1 ไม่เหมือน n == 1

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 14

โปรแกรมแสดงชื่อเดือน

- รับหมายเลขเดือน (มีค่าได้ตั้งแต่ 1 ถึง 12)
- แสดงชื่อเดือนตามหมายเลขที่ได้รับ
- ให้ m แทนหมายเลขเดือนที่ได้รับ
- สตริงย่อตัวตั้งแต่ตัวอักษรตัวที่ 9(m-1) ถึง 9m-1 ก็คือชื่อเดือน
- ถ้าผู้ใช้ป้อนเดือนที่ 2 ก็หยิบสตริงย่ออย่างล่างนี้

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
J	A	N	U	A	R	Y		F	E	B	R	U	A	R	Y	M	A	R	C	H		...	

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 16

โปรแกรมแสดงชื่อเดือน

- จะเกิดข้อผิดพลาดใดเมื่อป้อนเลขเดือนที่ 100

```
import jlab.JLabIO;
public class Month {
    public static void main(String [] args) {
        Enter month number : 100
        java.lang.StringIndexOutOfBoundsException: String index out
        of range: 900
            at java.lang.String.substring(String.java:1473)
            at Month.main(Month.java:10)
        Exception in thread "main"
        JLab>

        int m = JLabIO.readInt("Enter month number : ");
        System.out.println(month.substring(9*(m-1) , 9*m));
    }
}
```

เกิด **StringIndexOutOfBoundsException**

โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;

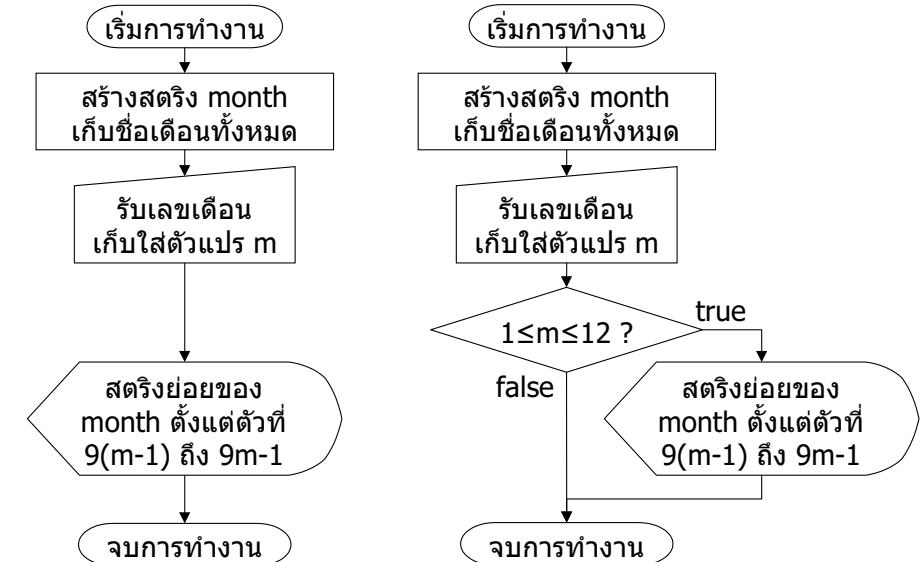
public class Month {
    public static void main(String [] args) {
        String month = "";
        month += "JANUARY FEBRUARY MARCH APRIL ";
        month += "MAY JUNE JULY AUGUST ";
        month += "SEPTEMBER OCTOBER NOVEMBER DECEMBER ";
        int m = JLabIO.readInt("Enter month number : ");
        if ((1 <= m) && (m <= 12)) {
            System.out.println(month.substring(9*(m-1) , 9*m));
        }
    }
}
```

1≤m≤12

→

(1 <= m) && (m <= 12)

การตรวจสอบเงื่อนไข



เงื่อนไขหนึ่ง เขียนได้หลายแบบ

- (1≤m) && (m≤12)
- (0<m) && (m<13)
- (12≥m) && (1≤m)
- (m==1)|||(m==2)|||(m==3)|||(m==4)|||(m==5)|||(m==6)|||(m==7)|||(m==8)|||(m==9)|||(m==10)|||(m==11)|||(m==12)
- ! (!(1≤m) || !(m≤12))
- ! ((m<1) || (m>12))

ตารางความจริง

- $(1 \leq m) \&\& (m \leq 12)$ คือ $!((m < 1) || (m > 12))$?

$(1 \leq m)$	$(m \leq 12)$	$(1 \leq m) \&\& (m \leq 12)$
true	true	true
true	false	false
false	true	false
false	false	false

$(m < 1)$	$(m > 12)$	$(m < 1) (m > 12)$	$!((m < 1) (m > 12))$
false	false	false	true
false	true	true	false
true	false	true	false
true	true	true	false

กฎการพิจารณาลำดับการปฏิบัติการ

- ภายในวงเล็บ
- unary + - ++ -- !
- type casting
- * / %
- binary + -
- < > <= >=
- == !=
- &&
- ||
- = *= /= %= += -=

```
int k;
...
boolean b = ! k==0; // wrong
...
if ( 3<=k+1 && k<=5 || k>=9 ) {
    // k=2,3,4,5, 9,10,11, ...
}
```

โปรแกรมแสดงจำนวนวันของเดือน ก.พ.

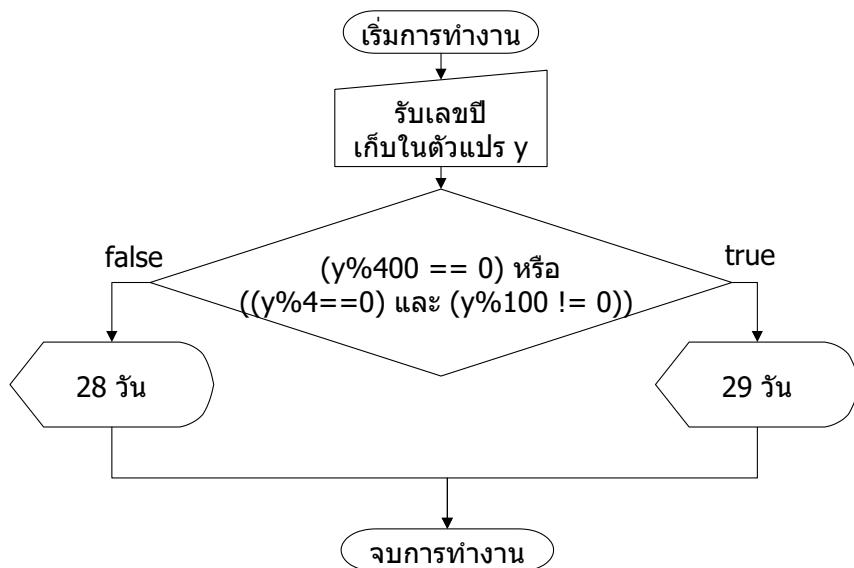
- Requirement
 - หาว่าเดือนก.พ. ของปีที่ให้มามี 28 หรือ 29 วัน
- Analysis
 - รับเลขปี (เป็น ค.ศ.) ผ่านทางแป้นพิมพ์
 - ใช้การทดสอบปีอุทกสุรทินเพื่อแสดงจำนวนวันในเดือน ก.พ.
- Design
 - เขียนผังงาน
- Implementation
 - เขียนโปรแกรม

การทดสอบปีอุทกสุรทิน

- เดือนกุมภาพันธ์ของปี y (ค.ศ.) มี 29 วันเมื่อ
 - เลขปี (ค.ศ.) ต้องหารด้วย 4 ลงตัว แต่ต้องหารด้วย 100 ไม่ลงตัว ยกเว้นกรณีที่หารด้วย 400 ลงตัว
 - ปี 2002 ก.พ. มี 28 วัน (หารด้วย 4 ไม่ลงตัว)
 - ปี 2004 ก.พ. มี 29 วัน (หารด้วย 4 ลงตัว และหารด้วย 100 ไม่ลง)
 - ปี 2100 ก.พ. มี 28 วัน (หารด้วย 100 ลงตัว)
 - ปี 2000 ก.พ. มี 29 วัน (หารด้วย 400 ลงตัว)
 - ให้ y คือเลขปี (ค.ศ.)
 - $(y \% 4) == 0$ แทนการทดสอบว่าหารด้วย 4 ลงตัว ?
 - $(y \% 100) != 0$ แทนการทดสอบว่าหารด้วย 100 ไม่ลงตัว ?
 - $(y \% 400) == 0$ แทนการทดสอบว่าหารด้วย 400 ลงตัว ?

```
if ( (y%4 == 4 && y%100 != 0) ||
     (y%400 == 0) )
```

ผังงาน



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 25

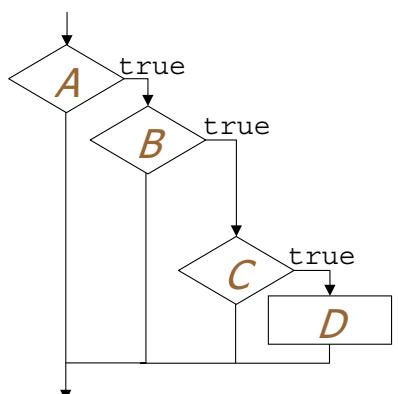
โปรแกรมแสดงจำนวนวันของเดือน ก.พ.

```
import jlab.JLabIO;
public class Feb {
    public static void main(String [] args) {
        int y = JLabIO.readInt("year = ");
        if ((y%4==0 && y%100!=0) || (y%400==0)) {
            System.out.println("Feb. has " + 29 + " days.");
        } else {
            System.out.println("Feb. has " + 28 + " days.");
        }
    }
}
int y = JLabIO.readInt("year = ");
int numDays;
if ((y%4==0 && y%100!=0) || (y%400==0)) {
    numDays = 29;
} else {
    numDays = 28;
}
System.out.println("Feb. has " + numDays + " days.");
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 26

Nested If

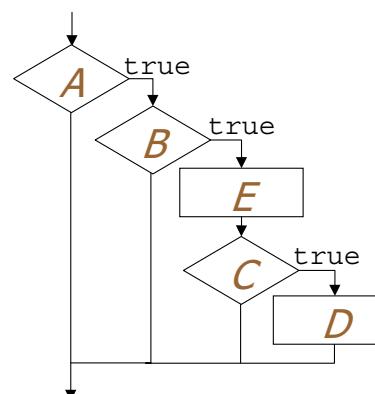


```
if ( A ) {
    if ( B ) {
        if ( C ) {
            D ;
        }
    }
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 27

Nested If

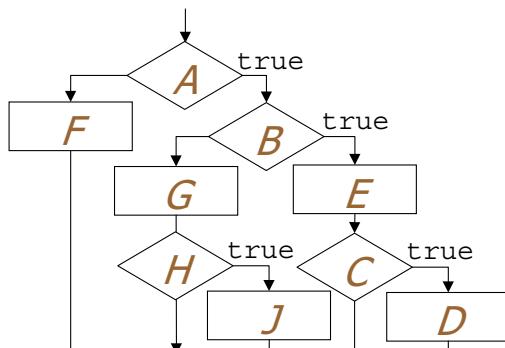


```
if ( A ) {
    if ( B ) {
        E ;
        if ( C ) {
            D ;
        }
    }
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

20/5/02 28

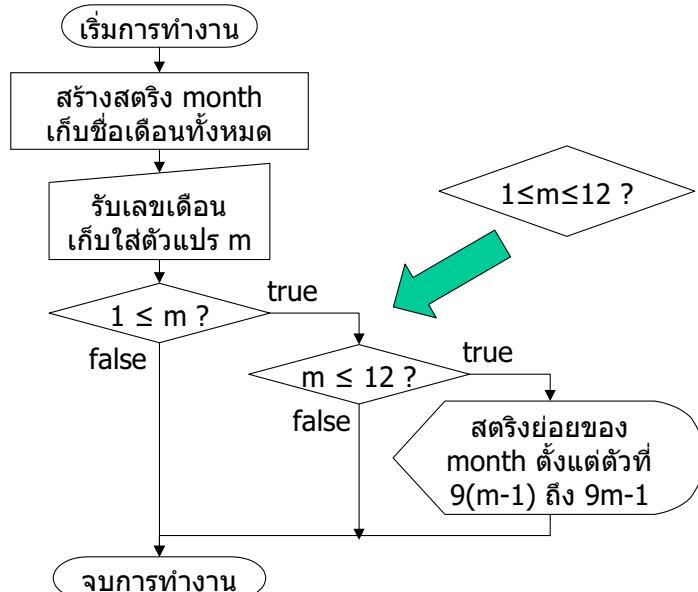
Nested If



```

if ( A ) {
    if ( B ) {
        E ;
        if ( C ) {
            D ;
        }
    } else {
        G ;
        if ( H ) {
            J ;
        }
    } else {
        F ;
    }
}
  
```

ผังงานแสดงชื่อเดือน



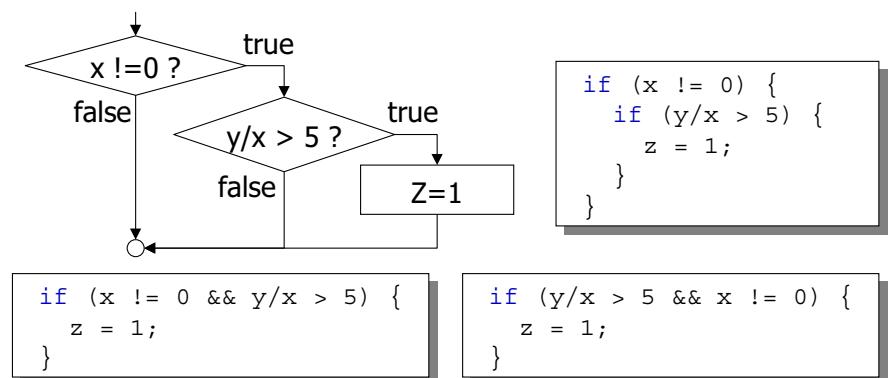
โปรแกรมแสดงชื่อเดือน

```

import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "";
        month += "JANUARY FEBRUARY MARCH APRIL ";
        month += "MAY JUNE JULY AUGUST ";
        month += "SEPTEMBER OCTOBER NOVEMBER DECEMBER ";
        int m = JLabIO.readInt("Enter month number : ");
        if (1 <= m) {
            if (m <= 12) {
                System.out.println(month.substring(9*(m-1), 9*m));
            }
        }
    }
}
  
```

Short-circuiting

ถ้า $x == 0$ จะไม่มีการคำนวณ y/x คำนวณ y/x ก่อนทดสอบ $x != 0$

Short-circuiting ("ไม่คิดหมวด พยายามหาทางลัด")

- กรณี $\&\&$ จะเลิกประเมิน expr ทางขวาถ้าไปถ้าพบ expr ที่เป็น false
- กรณี $\|$ จะเลิกประเมิน expr ทางขวาถ้าไปถ้าพบ expr ที่เป็น true

Statement และ Block

- Block คือกลุ่มของบรรทัดคำสั่ง ครอบด้วย { }

```
if (Boolean Expression)
    a statement ;
```

```
if (Boolean Expression)
    a statement ;
else
    a statement ;
```

```
if (Boolean Expression) {
    statements ;
}
```

```
if (Boolean Expression) {
    statements ;
} else {
    statements ;
}
```

if statement

if-else statement

เขียนได้หลายแบบ

if (x < 0) { x = -x; }	if (x < 0) x = -x;	if (x < 0) x = -x;
----------------------------	------------------------	--------------------

if (x < 0) { done = true; } else { done = false; }	if (x < 0) done = true; else done = false;	done = (x < 0);
--	--	-----------------

โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "Illegal month number";
        int m = JLabIO.readInt("Enter month number : ");
        if (m == 1) month = "JANUARY";
        if (m == 2) month = "FEBRUARY";
        if (m == 3) month = "MARCH";
        if (m == 4) month = "APRIL";
        if (m == 5) month = "MAY";
        if (m == 6) month = "JUNE";
        if (m == 7) month = "JULY";
        if (m == 8) month = "AUGUST";
        if (m == 9) month = "SEPTEMBER";
        if (m == 10) month = "OCTOBER";
        if (m == 11) month = "NOVEMBER";
        if (m == 12) month = "DECEMBER";
        System.out.println(month);
    }
}
```

โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;

public class Month {
    public static void main(String [] args) {
        String month = "Illegal month number";
        int m = JLabIO.readInt("Enter month number : ");
        if (m == 1) month = "JANUARY";
        else if (m == 2) month = "FEBRUARY";
        else if (m == 3) month = "MARCH";
        else if (m == 4) month = "APRIL";
        else if (m == 5) month = "MAY";
        else if (m == 6) month = "JUNE";
        else if (m == 7) month = "JULY";
        else if (m == 8) month = "AUGUST";
        else if (m == 9) month = "SEPTEMBER";
        else if (m == 10) month = "OCTOBER";
        else if (m == 11) month = "NOVEMBER";
        else if (m == 12) month = "DECEMBER";
        System.out.println(month);
    }
}
```

switch - case

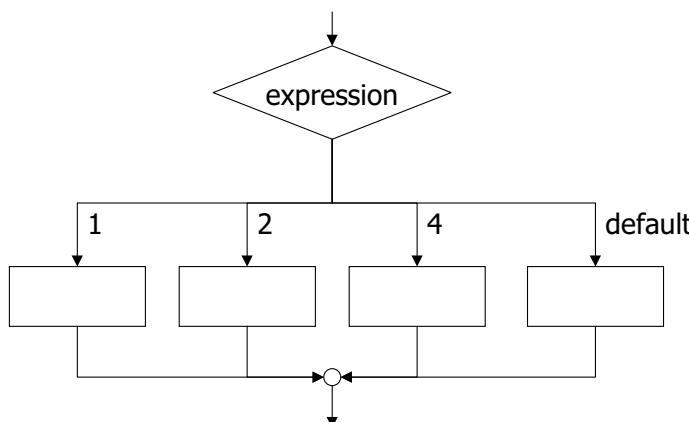
```
import jlab.JLabIO;
public class Month {
    public static void main(String [] args) {
        int m = JLabIO.readInt("Enter month number : ");
        switch (m) {
            case 1 : System.out.println("JANUARY"); break;
            case 2 : System.out.println("FEBRUARY"); break;
            case 3 : System.out.println("MARCH"); break;
            case 4 : System.out.println("APRIL"); break;
            case 5 : System.out.println("MAY"); break;
            case 6 : System.out.println("JUNE"); break;
            case 7 : System.out.println("JULY"); break;
            case 8 : System.out.println("AUGUST"); break;
            case 9 : System.out.println("SEPTEMBER"); break;
            case 10 : System.out.println("OCTOBER"); break;
            case 11 : System.out.println("NOVEMBER"); break;
            case 12 : System.out.println("DECEMBER"); break;
            default : System.out.println("Illegal month number");
        }
    }
}
```

switch-case

- ในวงเล็บหลัง **switch** ต้องเป็น expr ที่ได้ผลเป็นจำนวนเต็ม
- ตามหลัง **case** ต้องเป็นค่าคงตัวแบบจำนวนเต็ม
- ท้ายสุดของ **case** ถ้ามี **break** จะกระโดดออกนอก block
- ถ้าท้ายสุดของ **case** ไม่มี **break** การทำงานจะไหลต่อลงไปทำงานใน **case** ถัดลงไป
- ถ้าไม่ตรง **case** ได้เลย การทำงานจะไปที่ **default**

```
switch (m) {
    case 1 :
    case 2 :
        System.out.println("m = 1 or 2");
        break;
    case 4 :
        System.out.println("m = 4");
        break;
    default :
        System.out.println("m != 1, 2, and 4");
}
```

switch-case



สิ่งที่พบว่าผิดกันบ่อย

Left side: Common mistakes in switch-case statements.

- `if (0 <= x <= 10)`
`if x == 0`
`if (x >= 0 && <= 10)`
`if (x == 0 || == 1)`
`if (x = 0)`
`if (x => 0)`
- `if (x == 0)`
`if (y == 0) z = 0;`
`else if (a == 1)`
`z = 1;`
- `if (x != 0 || x != 5)`
`if (x < 3 && x > 9)`
`if (done && !done)`
`if (done || !done)`

Right side: Corrected versions of the mistakes.

- `if (0 <= x && x <= 10)`
`if (x == 0)`
`if (x >= 0 && x <= 10)`
`if (x == 0 || x == 1)`
`if (x == 0)`
`if (x >= 0)`
- `if (x == 0) {`
`if (y == 0) z = 0;`
`} else if (a == 1)`
`z = 1;`
- `if (x != 0 || x != 5)`
`if (x < 3 && x > 9)`
`if (done && !done)`
`if (done || !done)`

Bottom center: Text indicating the changes made.

เปลี่ยนไปไร้ค่า

อย่าใช้ == เพื่อเปรียบเทียบ floating point

- การเก็บจำนวนจริงด้วย `float` และ `double` มีข้อจำกัดเรื่องความแม่นยำ
- การคำนวณจึงมักทำให้ผลลัพธ์มีความคลาดเคลื่อน
- ไม่ควรนำจำนวนจริงมาทดสอบว่าเท่ากันหรือไม่

```
public class Imprecision {  
    public static void main(String [] args) {  
        double a = 49.0;  
  
        System.out.println( 1.0 == (1.0/a * a) );  
        System.out.println( 1.0 - (1.0/a * a) );  
    }  
}
```

JLab>java Imprecision
false
1.1102230246251565E-16
JLab>

การเปรียบเทียบจำนวนจริง x == y ?

- ตั้งค่า ϵ ไว้ที่ 10^{-14} ถ้าผ่านการทดสอบข้างบนนี้ ก็แสดงว่า x และ y มีค่าใกล้เคียงกันมาก

$$\frac{|x-y|}{\max(|x|, |y|)} \leq \epsilon$$

```
public class Epsilon {  
    public static void main(String [] args) {  
        double EPSILON = 1.0E-14;  
        double a = 49.0;  
        double x = 1.0, y = (1.0/a * a);  
        boolean eq = (Math.abs(x-y) /  
                      Math.max(Math.abs(x), Math.abs(y)))  
                    <= EPSILON;  
        System.out.println( eq );  
    }  
}
```

JLab>java Epsilon
true
JLab>

การใช้ == < > เพื่อเปรียบเทียบ `char`

- ใช้ == < > เพื่อเปรียบเทียบ `char` ได้ตามลำดับตัวอักษร (lexicographic ordering)
- การเปรียบเทียบอักษรคือการเปรียบเทียบตามรหัส Unicode ที่ใช้แทนอักษรนั้น

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
002	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
003	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005	P	Q	R	S	T	U	V	W	X	Y	Z	[\	^	_	
006	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

อย่าใช้ == < > เพื่อเปรียบเทียบ `String`

- `String` ไม่ใช้ประเภทพื้นฐาน
- ต้องใช้ methods `equals` และ `compareTo` ในการเปรียบเทียบ

$$s1.equals(s2) = \begin{cases} \text{true} & \text{ถ้า } s1 \text{ เหมือนกับ } s2 \\ \text{false} & \text{ถ้า } s1 \text{ ไม่เหมือนกับ } s2 \end{cases}$$
$$s1.compareTo(s2) = \begin{cases} < 0 & \text{ถ้า } s1 \text{ มาก่อน } s2 \text{ ตามลำดับตัวอักษร} \\ = 0 & \text{ถ้า } s1 \text{ เหมือนกับ } s2 \\ > 0 & \text{ถ้า } s1 \text{ มากหลัง } s2 \text{ ตามลำดับตัวอักษร} \end{cases}$$

การเปรียบเทียบ String

```
public class StringCompare {  
    public static void main(String [] args) {  
        System.out.println("Computer".equals("Computer"));  
        System.out.println("Computer".compareTo("Computer"));  
        System.out.println("Computer".compareTo("Engineer"));  
        System.out.println("Computer".compareTo("Comp"));  
        System.out.println("Computer".compareTo("Computing"));  
    }  
}
```

```
JLab>java StringCompare  
true  
0  
-2  
4  
-4  
JLab>
```

2110101 : การทำโปรแกรมคอมพิวเตอร์

Iteration : while, do-while, for

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

โปรแกรมหาค่าเฉลี่ย

```
import jlab.JLabIO;

public class Mean {
    public static void main(String [] args) {
        double s = 0.0;
        s += JLabIO.readDouble("data #1 : ");
        s += JLabIO.readDouble("data #2 : ");
        s += JLabIO.readDouble("data #3 : ");
        s += JLabIO.readDouble("data #4 : ");
        s += JLabIO.readDouble("data #5 : ");
        System.out.println("mean = " + (s / 5.0));
    }
}
```

ถ้าต้องการหาค่าเฉลี่ย
ของข้อมูล 50 จำนวน
จะเขียนอย่างไรส้นๆ

```
JLab>java Mean
data #1 : 12
data #2 : 35.5
data #3 : 16.8
data #4 : 19.0
data #5 : 2.2
mean = 17.1
JLab>
```

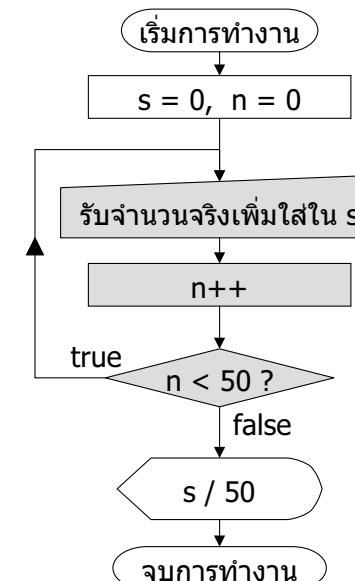
หัวข้อ

- คำสั่ง do-while
- คำสั่ง while
- คำสั่ง for
- การอ่านแฟ้มข้อมูลแบบลำดับ
- คำสั่ง break และ continue

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 2

ผังงานแบบวน do-while



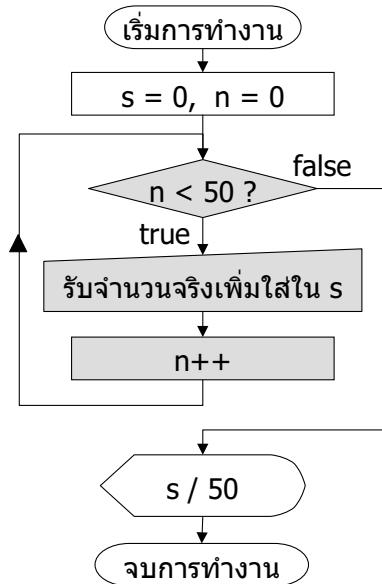
```
double s = 0.0;
int n = 0;

do {
    s += JLabIO.readDouble();
    n++;
}

} while( n < 50 );

System.out.println( s/50 );
```

ผังงานแบบวนวน while



```
double s = 0.0;
int n = 0;

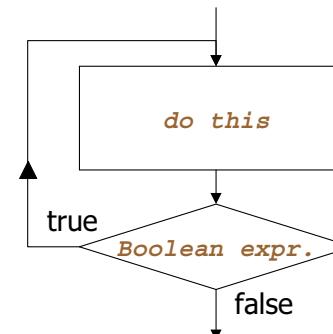
while ( n < 50 ) {
    s += JLabIO.readDouble();
    n++;
}

System.out.println( s/50 );
```

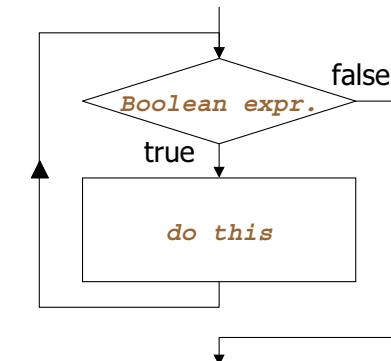
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 5

วงวนแบบ do-while และ while



```
do {
    do this
} while ( Boolean expr. );
```



```
while ( Boolean expr. ) {
    do this
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 6

โปรแกรมหารากที่สอง

- Requirement
 - อ่านจำนวนจากผู้ใช้ และแสดงรากที่สองของจำนวนนั้น
- Analysis
 - รับจำนวนจริงผ่านทางแป้นพิมพ์
 - ใช้สูตรในการหารากที่สอง $x_{k+1} = \frac{1}{2} \left(x_k + \frac{a}{x_k} \right)$
 - แสดงรากที่สองที่หาได้ทางจอภาพ
- Design
 - เขียนผังงาน
- Implementation
 - เขียนโปรแกรม

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 7

การประมาณค่ารากที่สอง

- ต้องการหารากที่สองของ a
- ถ้า x_k เป็นค่าประมาณของ \sqrt{a}
- ค่าของ $x_{k+1} = (x_k + a/x_k)/2$ จะเป็นค่าประมาณที่ใกล้ \sqrt{a} มากกว่า x_k (ไม่พิสูจน์)

```
a = 4.0
x0 = 1.0
x1 = 2.5
x2 = 2.05
x3 = 2.000609756097561
x4 = 2.000000929222947
x5 = 2.0000000000000002
x6 = 2.0
```

```
a = 100.0
x0 = 1.0
x1 = 50.5
x2 = 26.24009900990099
x3 = 15.025530119986813
x4 = 10.840434673026925
x5 = 10.032578510960604
x6 = 10.000052895642693
x7 = 10.000000000139897
x8 = 10.0
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 8

แล้วจะประมาณค่าไปสักกี่รอบ ?

- หาค่า $x_{k+1} = (x_k + a/x_k)/2$ $k = 1,2,3,\dots$
เริ่มด้วย $x_0 = 1$
- นำไปจน x_{k+1} กับ x_k มีค่าใกล้เคียงกันมาก
- x และ y (แบบ double) มีค่าใกล้เคียงกันมากเมื่อ

$$\frac{|x - y|}{\max(|x|, |y|)} \leq \varepsilon$$

(ให้ ε มีค่าเท่ากับ 10^{-14})

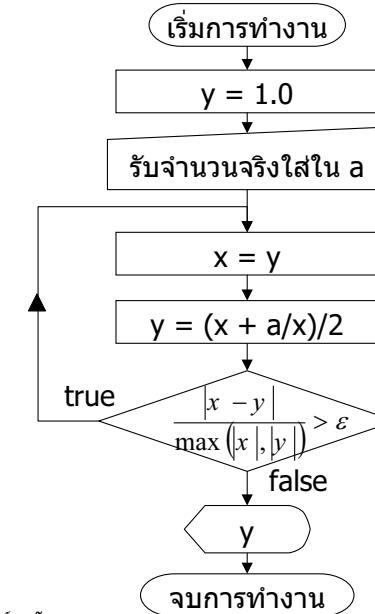
โปรแกรมหารากที่สอง

```
import jlab.JLabIO;
public class Sqrt {
    public static void main(String[] args) {
        double x, y = 1.0;
        double a = JLabIO.readDouble("a = ");
        do {
            x = y;
            y = (x + a / x) / 2.0;
        } while ((Math.abs(x - y) /
                  Math.max(Math.abs(x), Math.abs(y))) 
                  >= 1E-14);
        System.out.println("sqrt(" + a + ") = " + y);
    }
}
```

```
JLab>java Sqrt
x = 9E200
sqrt(9.0E200) = 3.0E100
JLab>
```

หมุนในวงวน 339 รอบ

ผังงานการหารากที่สอง



การอ่านแฟ้มข้อความขั้นพื้นฐาน

- แฟ้ม (file) คือที่เก็บข้อมูลในหน่วยความจำสำรอง (โดยทั่วไปคือ harddisk)
- แต่ละแฟ้มมีชื่อกำกับ
- Text file คือแฟ้มข้อความที่เก็บสตริง เป็นบรรทัดๆ แต่ละบรรทัดคั่นด้วยรหัสขึ้นบรรทัดใหม่

C:\DATA\SCORE.TXT

4531001321	3
4531003621	3
4531004221	4
4531005921	3
4531006521	3
4531101421	4
4531102021	4
4531102221	0

การใช้ JLabIO อ่านไฟล์ข้อความ

```
import java.io.*;
import jlab.JLabIO;

// open file
BufferedReader in = JLabIO.openFile("c:\dat.txt");
...
// read the next line from file
String txt = in.readLine();
if (txt == null) // end of file
...
// close the file
in.close();
```

โปรแกรมแสดง text file อุகจกภาพ

```
import jlab.JLabIO;
import java.io.*;

public class DumpFile1 {
    public static void main(String[] args)
        throws IOException {

        String fn = JLabIO.readString("file name : ");
        BufferedReader in = JLabIO.openFile(fn);
        String txt;
        do {
            txt = in.readLine();
            if (txt != null) System.out.println(txt);
        } while ( txt != null );
        in.close();
    }
}
```

โปรแกรมแสดง text file อุกจกภาพ

```
import jlab.JLabIO;
import java.io.*;

public class DumpFile2 {
    public static void main(String[] args)
        throws IOException {

        String fn = JLabIO.readString("file name : ");
        BufferedReader in = JLabIO.openFile(fn);
        String txt;
        while ( (txt = in.readLine()) != null ) {
            System.out.println(txt);
        }
        in.close();
    }
}
```

ทำไมต้องมี throws IOException ที่ main ?

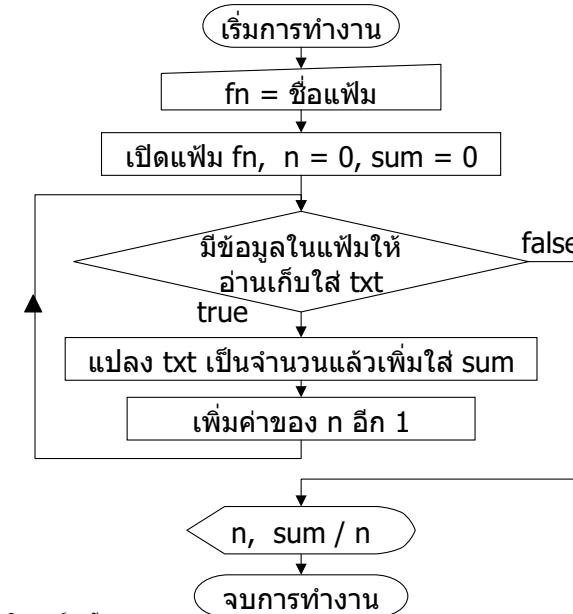
- มีโอกาสเกิดข้อผิดพลาดระหว่างการทำงาน ซึ่งเกี่ยวข้องกับแฟ้มข้อมูลได้หลายประการ (IOException)
 - เปิดแฟ้มที่ไม่มีอยู่จริง (เช่นสะกดชื่อผิด)
 - ตอนอ่านข้อมูลจากแฟ้ม อาจเกิดข้อผิดพลาดขึ้นได้ (เช่น อ่านจาก diskette แต่ผู้ใช้ดันดึงแผ่นออก)
- เพื่อความง่าย ขอไม่สนใจข้อผิดพลาดเหล่านี้
- เลยอนข้อผิดพลาดแบบ IOException ให้ระบบ (ตัวโปรแกรมที่เราเขียนไม่ขอรับผิดชอบ)

IOException เป็นชื่อ class มาตรฐานในระบบจาวา

โปรแกรมหาค่าเฉลี่ย

- Requirement
 - หาค่าเฉลี่ยจากข้อมูลทั้งหมดในไฟล์
- Analysis
 - ผู้ใช้ระบุชื่อไฟล์ทางแป้นพิมพ์
 - ไฟล์ข้อมูลเก็บจำนวนจริงบรรทัดละหนึ่งจำนวน
 - ใช้ `Double.parseDouble(s)` แปลงสตริง s เป็น double
 - แสดงจำนวนข้อมูล และค่าเฉลี่ยที่หาได้ทางจอภาพ
- Design
 - เขียนผังงาน
- Implementation
 - เขียนโปรแกรม

ผังงานการหาค่าเฉลี่ย



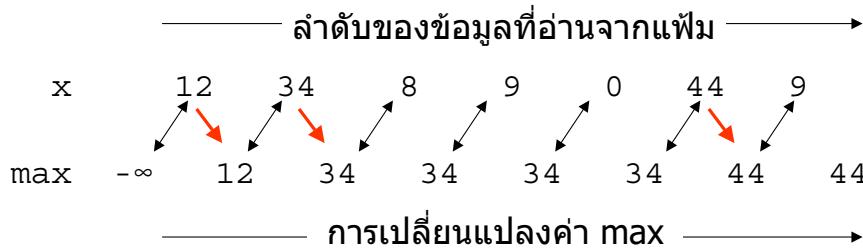
โปรแกรมหาค่าเฉลี่ย

```
import jlab.JLabIO;
import java.io.*;
public class Average1 {
    public static void main(String[] args)
        throws IOException {
        String fn = JLabIO.readString("file name : ");
        BufferedReader in = JLabIO.openFile(fn);
        int n = 0;
        double sum = 0.0;
        while ((txt = in.readLine()) != null) {
            sum += Double.parseDouble(txt);
            n++;
        }
        in.close();
        System.out.println("n = " + n +
                           " avg = " + sum / n);
    }
}
```

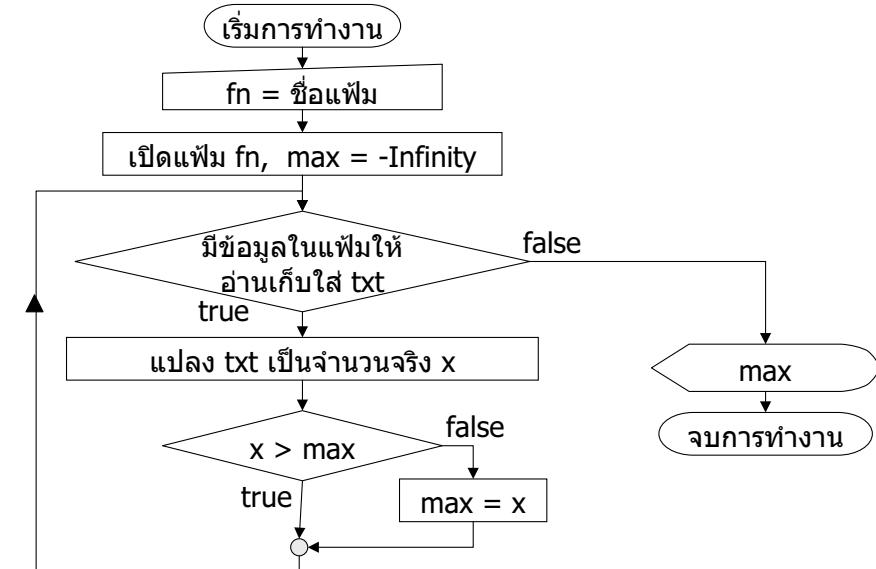
โปรแกรมหาค่ามากสุด

- Requirement
 - หาค่ามากสุดจากข้อมูลทั้งหมดในไฟล์
- Analysis
 - ผู้ใช้ระบุชื่อไฟล์ทางแป้นพิมพ์
 - ไฟล์ข้อมูลเก็บจำนวนจริงบรรทัดละหนึ่งจำนวน
 - แสดงค่ามากสุดที่หาได้ทางจอภาพ
- Design
 - เขียนผังงาน
- Implementation
 - เขียนโปรแกรม

การหาค่ามากสุด



ผังงานการหาค่ามากสุด



โปรแกรมหาค่ามากสุด

```
import jlab.JLabIO;
import java.io.*;
public class MaxValue {
    public static void main(String[] args)
        throws IOException {

        String fn = JLabIO.readString("file name : ");
        BufferedReader in = JLabIO.openFile(fn);
        double x, max = Double.NEGATIVE_INFINITY;
        while ((txt = in.readLine()) != null) {
            x = Double.parseDouble(txt);
            if (x > max) max = x;
        }
        in.close();
        System.out.println("max = " + max);
    }
}
```

การใช้ while กับโปรแกรม EAN-13

```
import jlab.JLabIO;
public class EAN13CheckDigit {
    public static void main(String[] args) {
        int s = 0;
        String d = JLabIO.readString("Enter 12 digits : ");
        s += Character.digit(d.charAt(1), 10);
        s += Character.digit(d.charAt(3), 10);
        s += Character.digit(d.charAt(5), 10);
        s += Character.digit(d.charAt(7), 10);
        s += Character.digit(d.charAt(9), 10);
        s += Character.digit(d.charAt(11), 10);
        s *= 3;
        s += Character.digit(d.charAt(0), 10);
        s += Character.digit(d.charAt(2), 10);
        s += Character.digit(d.charAt(4), 10);
        s += Character.digit(d.charAt(6), 10);
        s += Character.digit(d.charAt(8), 10);
        s += Character.digit(d.charAt(10), 10);
        System.out.println("Check Digit is " + (10 - (s%10))%10);
    }
}
```

การใช้ while กับโปรแกรม EAN-13

```
import jlab.JLabIO;
public class EAN13CheckDigit {
    public static void main(String[] args) {
        int s = 0, k;
        String d = JLabIO.readString("Enter 12 digits : ");

        k = 1;
        while (k<12) {
            s += Character.digit(d.charAt(k), 10);
            k += 2;
        }
        s *= 3;

        k = 0;
        while (k<12) {
            s += Character.digit(d.charAt(k), 10);
            k += 2;
        }
        System.out.println("Check Digit is " + (10-(s%10))%10);
    }
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 25

การใช้ while กับโปรแกรม EAN-13

```
import jlab.JLabIO;
public class EAN13CheckDigit {
    public static void main(String[] args) {
        int s = 0, k;
        String d = JLabIO.readString("Enter 12 digits : ");

        k = 0;
        while (k<12) {
            if ( (k % 2) == 0 ) {
                s += Character.digit(d.charAt(k), 10);
            } else {
                s += 3 * Character.digit(d.charAt(k), 10);
            }
            k++;
        }

        System.out.println("Check Digit is " + (10-(s%10))%10);
    }
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 26

while และ do-while

- วงวนที่เขียนด้วย while เปลี่ยนเป็น do-while ได้
- วงวนที่เขียนด้วย do while เปลี่ยนเป็น while ได้
- โปรแกรมเมอร์ควรเลือกแบบที่สื่อความหมาย อ่านง่าย
- while : ตราชบท่าที่เงื่อนไขเป็นจริง จึงจะทำ
 - อาจไม่ได้ทำใน block
- do-while ทำไปเรื่อยๆ ตราชบท่าที่เงื่อนไขยังเป็นจริง
 - อย่างน้อยต้องทำใน block หนึ่งครั้ง

วงวนที่ทำเป็นจำนวนครั้งตามที่กำหนด

- มีตัวแปรค่อนบันตามจำนวนรอบ

```
i = 1;
while (i <= n) {

    // do something here

    i++;
}
```

```
i = 0;
while (i < n) {

    // do something here

    i++;
}
```

```
i = n;
while (i > 0) {

    // do something here

    i--;
}
```

```
i = n-1;
while (i >= 0) {

    // do something here

    i--;
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 27

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 28

ข้อควรระวัง

- วนขาดหนึ่งรอบ วนเกินหนึ่งรอบ หรือวนติด loop

```
i = 1;  
while (i < n) {  
  
    // do something here  
  
    i++;  
}
```

```
i = 0;  
while (i <= n) {  
  
    // do something here  
  
    i++;  
}
```

```
i = n;  
while (i > 1) {  
  
    // do something here  
  
    i--;  
}
```

```
i = 0;  
while (i != n) {  
    // The test above can  
    // cause infinite  
    // loop.  
    i++;  
}
```

วงวนแบบ for

- วงวนที่ทำเป็นจำนวนครั้งตามที่กำหนด (counting loop) เป็นวงวนที่มักพบบ่อยมาก
- ตัวภาษาจึงมีคำสั่ง for เพื่อทำ counting loop

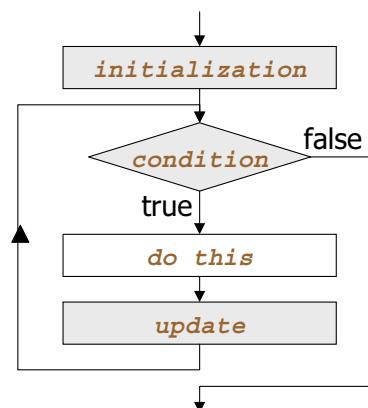
```
i = 1;  
while (i <= n) {  
    // do something here  
    i++;  
}
```

```
for (i = 1; i <= n; i++) {  
    // do something here  
}
```

```
i = n;  
while (i > 0) {  
    // do something here  
    i--;  
}
```

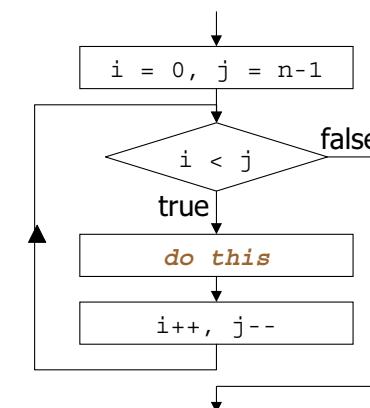
```
for (i = n; i > 0; i--) {  
    // do something here  
}
```

วงวนแบบ for



```
for ( initialization ; condition ; update ) {  
    do this  
}
```

comma



```
for ( i = 0, j = n-1; i < j; i++, j-- ) {  
    do this  
}
```

ตัวอย่างวงวนแบบ for

```
for (i = 1; i <= n; i++) {  
    System.out.println(i);  
}
```

```
for (i = 1; i <= n; i += 2) {  
    System.out.println(i);  
}
```

```
n = JLabIO.readInt("n = ");  
for (i = 2; (i < n) && (n % i != 0); i++) {  
}  
if (i < n) {  
    System.out.println(n + " = " + i + "x" + (n/i));  
} else {  
    System.out.println(n + " is prime.");  
}
```

อย่าใช้ for ผิดวัตถุประสงค์

```
double x = 0, y = 1;  
double a = JLabIO.readDouble("a = ");  
while ((Math.abs(x - y) /  
        Math.max(Math.abs(x), Math.abs(y)))  
       >= 1E-14) {  
    x = y;  
    y = (x + a / x) / 2.0;  
}  
System.out.println("sqrt(" + a + ") = " + y);
```

```
double x = 0, y = 1;  
double a = JLabIO.readDouble("a = ");  
for ( ;  
     (Math.abs(x - y) /  
      Math.max(Math.abs(x), Math.abs(y))) >= 1E-14);  
    x = y, y = (x + a / x) / 2.0 );  
System.out.println("sqrt(" + a + ") = " + y);
```

หนึ่ง block หรือ หนึ่ง statement

```
for ( initialization ; condition ; update ) {  
    statements  
}
```

```
for ( initialization ; condition ; update )  
    statement ;
```

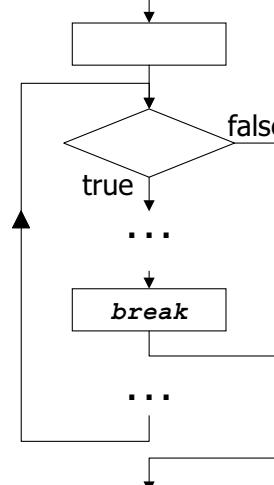
```
do {  
    statements  
} while ( Boolean expr );
```

```
while ( Boolean expr ) {  
    statements  
}
```

```
do  
    statement ;  
while ( Boolean expr );
```

```
while ( Boolean expr )  
    statement ;
```

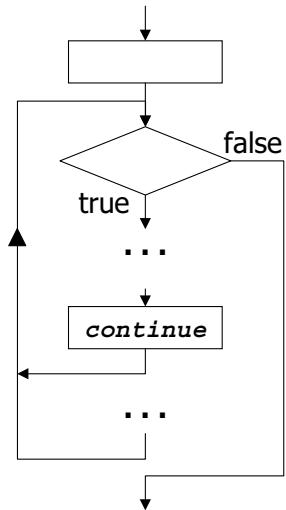
คำสั่ง break เพื่อออกจากรวงวน



```
while ((t = in.readLine()) != null) {  
    if ("END".equals(t)) break;  
    x = Double.parseDouble(t);  
    if (x > max) max = x;  
}
```

สนใจหาค่ามากสุดของจำนวนที่เก็บ
ในแฟ้มข้อมูลก่อนบรรทัดที่มีคำว่า
END

คำสั่ง continue เพื่อกลับไปต้นวงวน

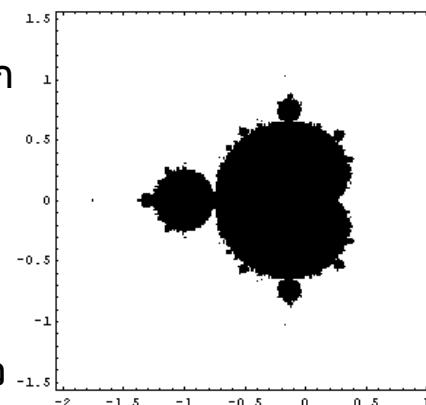


```
while ((t = in.readLine()) != null) {  
    x = Double.parseDouble(t);  
    if (x <= 0) continue;  
    if (x > max) max = x;  
}
```

สนใจหาค่ามากสุดของเฉพาะจำนวนเต็มบวก ที่เก็บในแฟ้มข้อมูล

Mandelbrot Set

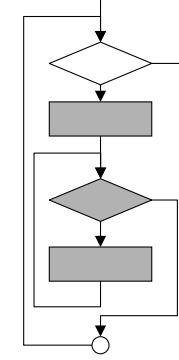
- ให้ลำดับของจำนวน เชิงซ้อน d_n คำนวณได้จาก $d_n = (d_{n-1})^2 + c$ โดยที่ $d_0 = 0+i0 = 0$ และ c เป็นจำนวนเชิงซ้อน
- c เป็นสมาชิกของ Mandelbrot set ถ้าลำดับของ $|d_n|$ ล่าช้าหากค่าคงตัว



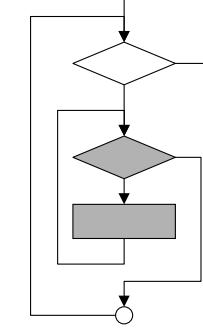
แกน x แทนเส้นจำนวนจริง แกน y แทนเส้นจำนวนจินตภาพ จุดดำเนินรูปคือสมาชิกของ Mandelbrot set

Nested Loops

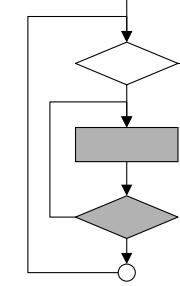
- ภายในวงวนหนึ่งจะมีวงวนอีกได้



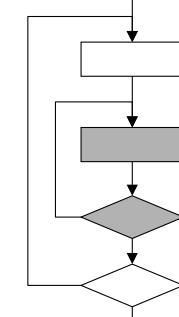
for ใน while



while ใน while



do-while ใน while



do-while ใน do-while

break และ continue ภายในวงวนใด ก็จะทำงานกับวงวนนั้น

การบวกและคูณจำนวนเชิงซ้อน

- $| (x+iy) | = (x^2 + y^2)^{1/2}$
- $(x+iy)+(p+iq) = (x+p) + i(y+q)$
- $(x+iy)(p+iq) = (xp-yq) + i(xq+yp)$

ตัวอย่างการเปลี่ยนแปลงของค่า r

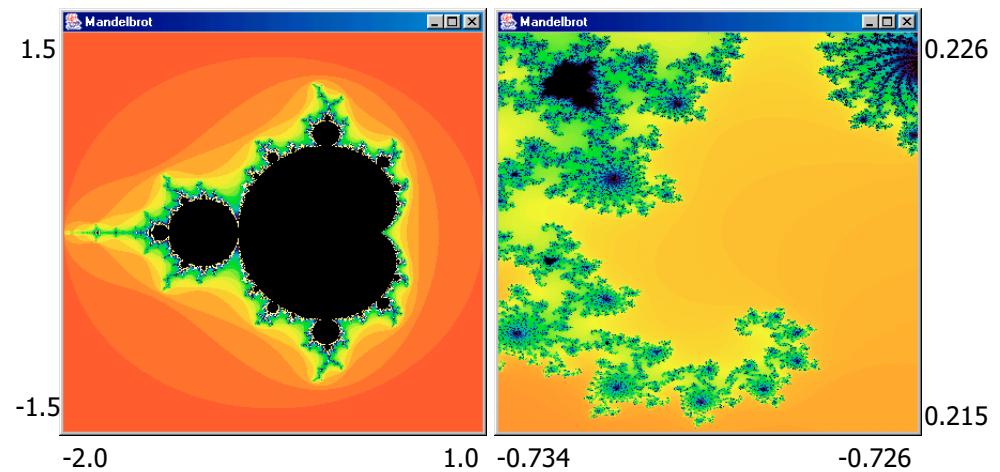
$c = (0.2+i0.2)$ จะได้
0.344, 0.351, 0.327, 0.305, 0.304, 0.312, 0.316, 0.316, 0.314,
0.313, 0.313, 0.313, 0.314, 0.314, 0.314, ... (converge)

$c = (1.0+i1.0)$ จะได้
3.162, 9.899, 97.01, 9409, 8.853E7, 7.837E15, 6.142E31,
3.773E63, 1.424E127, ... (diverge to infinity)

โปรแกรมแสดง Mandelbrot set

```
public void paint(Graphics g) {
    double xnew, ynew, xold, yold, x, y;
    double stepX = 3.0 / getSize().width;
    double stepY = 3.0 / getSize().height;
    int n;
    for (y = -1.5; y <= 1.5; y += stepY) {
        for (x = -2.0; x <= 1.0; x += stepX) {
            for (xold = x, yold = y, n = 0; (n < 256); n++) {
                xnew = xold * xold - yold * yold + x;
                ynew = 2 * xold * yold + y;
                if (Math.sqrt(xnew * xnew + ynew * ynew) > 2) break;
                xold = xnew;
                yold = ynew;
            }
            if (n >= 256) {
                g.drawRect((int)((x + 2) / stepX),
                           (int)((y + 1.5) / stepY), 1, 1);
            }
        }
    }
}
```

Visualizing Mandelbrot Set



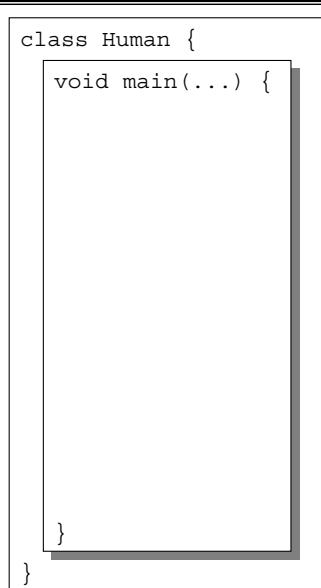
2110101 : การทำโปรแกรมคอมพิวเตอร์

Methods

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

การเขียนโปรแกรม

- ปัญหาของโปรแกรมภาษาฯ
 - อ่านยาก
 - debug ยุ่ง
 - แก้ไขปรับปรุงลำบาก
- ข้อสังเกตของโปรแกรมทั่วไป
 - สามารถแบ่งการทำงานออกเป็นส่วนๆ ตามหน้าที่
 - มีการทำงานช้าๆ บ่อยๆ

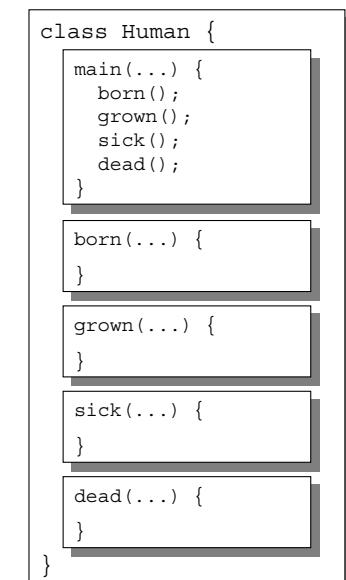


หัวข้อ

- บทบาทของ methods
- Methods ต่างๆ ของ class Math
- องค์ประกอบของ method
- การเรียกใช้
- Overloading

การเขียนโปรแกรม

- ทางแก้
 - แบ่งโปรแกรมเป็นส่วนย่อยๆ
 - แยกส่วนของโปรแกรมที่ซ้ำๆ คล้ายๆ กันออกมานำเสนอ
- เรียกส่วนย่อยเหล่านี้ว่า subroutine, subprogram, หรือ method
- แต่ละ method มีหน้าที่ในตัวเองอย่างเด่นชัด



เมื่อต้องเขียนโปรแกรมใหม่

- คิดถึงโปรแกรมเก่าที่เคยเขียนไว้
- คิดถึงคลังคำสั่งของระบบที่มีให้ใช้ได้ (Java API)
- นำของที่มีอยู่มาแก้ไข (แฟ้ม .java)
- นำของที่มีอยู่มาใช้ใหม่ (แฟ้ม .class)
- เขียนคำสั่งเพื่อเชื่อมต่อของที่มีอยู่ให้ใช้ได้
- ถ้าจำเป็น จึงเขียนใหม่
 - เขียนให้อ่านง่าย (readability)
 - เขียนให้แก้ไขง่าย (maintainability)
 - เขียนให้สามารถนำไปใช้ในงานอื่นๆ ได้ในอนาคต (reuseability)

ตัวอย่าง method ที่เคยใช้

```
System.out.println("Hello World");
```

ส่ง "Hello World" เพื่อให้ System.out.println
(แสดงออกทางจอภาพ)

```
int x = JLabIO.readInt("x = ");
```

ส่ง "x = " เพื่อให้ JLabIO.readInt (แสดงออกทางจอภาพ
แล้วรับการป้อนจำนวนเต็มทางแป้นพิมพ์) และได้ผลลัพธ์เป็นจำนวน
เต็มคืนมา เก็บไว้ในตัวแปร x

```
double r = Math.random();
```

ไม่ส่งอะไรให้ Math.random แต่จะได้ผลลัพธ์เป็นจำนวนจริงคืนมา
(ซึ่งเป็นจำนวนจริงสุ่มในช่วง 0.0 ถึง 1.0) เก็บไว้ในตัวแปร x

Methods ต่างๆ ของคลาส Math

- abs(x) : absolute value
- acos(x) : arc cosine
- asin(x) : arc sine
- atan(x) : arc tangent
- cos(x) : trigonometric cosine
- sin(x) : trigonometric sine
- tan(x) : trigonometric tangent
- exp(x) : e^x
- log(x) : natural logarithm (base e)
- pow(x, y) : x^y

Methods ต่างๆ ของคลาส Math (ต่อ)

- ceil(x) : ceiling
- floor(x) : floor
- max(x, y) : the greater of x and y
- min(x , y) : the smaller of x and y
- random() : a random number in [0.0, 1.0)
- round(x) : the closest integer to x
- sqrt(x) : positive square root of x
- toDegrees(x) : convert angle in radian to degree
- toRadians(x) : convert angle in degree to radian

ตัวอย่าง Java API Help File

```
public static double sin(double a)
```

Returns the trigonometric sine of an angle. Special cases:

- If the argument is NaN or an infinity, then the result is NaN.
- If the argument is zero, then the result is a zero with the same sign as the argument.

A result must be within 1 ulp of the correctly rounded result. Results must be semi-monotonic.

Parameters:

a - an angle, in radians.

Returns:

the sine of the argument.

ulp - unit of the last place

แยกส่วนช้าๆ คล้ายๆ กันเป็นเมธ็อด

$$\cos(\theta) = \frac{u_X v_X + u_Y v_Y}{\sqrt{u_X^2 + u_Y^2} \sqrt{v_X^2 + v_Y^2}}$$

```
cosine = (ux * vx + uy * vy) /  
        (Math.sqrt(ux * ux + uy * uy) *  
         Math.sqrt(vx * vx + vy * vy));
```

```
cosine = (ux * vx + uy * vy) /  
        (length(ux, uy) * length(vx, vy));
```

```
public static double length(double x, double y) {  
    return Math.sqrt(x * x + y * y);  
}
```

โปรแกรมหานมระหว่างสองเวกเตอร์

Requirement

- หานม (เป็นองศา) ระหว่างสองเวกเตอร์ (u_X, u_Y) (v_X, v_Y)

Analysis

- เวกเตอร์มีจุดเริ่มที่ $(0, 0)$
- รับเวกเตอร์จากแป้นพิมพ์ โดยระบุตำแหน่งของจุดปลาย (u_X, u_Y) และ (v_X, v_Y)
- สูตรการหานมคือ

$$\theta = \cos^{-1} \left(\frac{u_X v_X + u_Y v_Y}{\sqrt{u_X^2 + u_Y^2} \sqrt{v_X^2 + v_Y^2}} \right)$$

- แสดงมุมที่หาได้ทางจอภาพ

คล้ายกันมาก

โปรแกรมหานมระหว่างสองเวกเตอร์

```
import jlab.JLabIO;  
public class A {  
    public static void main(String[] args) {  
        double ux = JLabIO.readDouble("ux = ");  
        double uy = JLabIO.readDouble("uy = ");  
        double vx = JLabIO.readDouble("vx = ");  
        double vy = JLabIO.readDouble("vy = ");  
  
        double cosine = (ux * vx + uy * vy) /  
                        (length(ux, uy) * length(vx, vy));  
  
        double degree = Math.toDegrees(Math.acos(cosine));  
        System.out.println(JLabIO.format(degree, 5, 2));  
    }  
  
    public static double length(double x, double y) {  
        return Math.sqrt(x * x + y * y);  
    }  
}
```

องค์ประกอบของเมท็อด

หัวเมท็อด

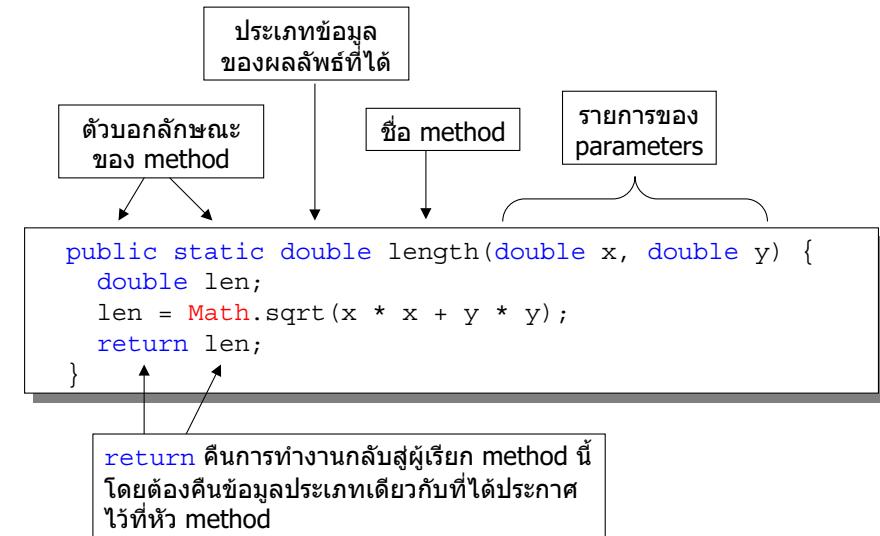
- บอกลักษณะการใช้งานของเมท็อด
- ระบุชื่อ ประเภทของข้อมูลที่รับ และประเภทของผลลัพธ์ที่ได้

ตัวเมท็อด

- ประกาศตัวแปรใช้ภายใน
- บรรยายการทำงานของเมท็อด
- คืนการทำงานสู่ผู้เรียก

```
public static double length(double x, double y) {  
    double len;  
    len = Math.sqrt(x * x + y * y);  
    return len;  
}
```

องค์ประกอบของเมท็อด



หัวเมท็อด : ประเภทข้อมูลของผลลัพธ์

- ประเภทข้อมูลพื้นฐาน (`int`, `double`, `boolean`, ...)
- ประเภทข้อมูลแบบคลาส (`String`, `Point`, ...)
- ถ้าเป็น method ที่ไม่คืนผลลัพธ์ ให้ใส่คำว่า `void`

```
public static int min(int a, int b)  
  
int m = min(a, 0);
```

```
public static String toString(int i, int radix)  
  
String s = toString(a, 16);
```

```
public static void gc()  
  
gc();
```

หัวเมท็อด : ชื่อ

- การตั้งชื่อใช้กฎเหมือนกับตัวแปร
 - ประกอบด้วยตัวอักษร ตัวเลข ตัว \$ หรือ _ ก็ได้
 - ห้ามขีดเดียวตัวเลข
 - ไม่มีข้อจำกัดเรื่องความยาวของชื่อ
 - ตัวอักษรตัวใหญ่ไม่เหมือนตัวเล็ก
 - ต้องไม่ซ้ำกับคำส่วนของภาษาจาวา
- โดยทั่วไปชื่อเมท็อดขึ้นต้นด้วยตัวเล็ก
- มักตั้งชื่อเมท็อดให้เป็นกริยา

```
fillCircle setEnabled turnLeft println start run
```

หัวเมธ็อด : รายการของพารามิเตอร์

• รายการของ parameters

- parameter เป็นตัวแปรสำหรับรับข้อมูลเข้ามาประมวลผล
- แต่ละ parameter เขียนเหมือนการประกาศตัวแปร (แต่ไม่ต้องปิดท้ายด้วย semi-colon)
- แต่ละ parameter คั่นด้วยจลภาค (comma)
- รายการของ parameters อุ่ภัยในวงเล็บ (ถ้าไม่รับ parameter ใดๆ ก็ไม่ต้องใส่อีกในวงเล็บ)

```
public static void turnLeft(double angle)
```

```
public static double pow(double a, double b)
```

```
public static String readString()
```

ตัวเมธ็อด

```
public static double readDouble(String msg) {  
  
    double x = 0; // local var. declaration  
    boolean success = false; // local var. declaration  
  
    do {  
        String s; // local var. declaration  
        s = readString(msg);  
        try {  
            x = Double.parseDouble(s);  
            success = true;  
        } catch (NumberFormatException e) {  
            System.err.println("invalid real number, try again");  
            success = false;  
        }  
    } while (!success);  
  
    return x; // return to caller  
}
```

ตัวเมธ็อด : การคืนการทำงานสู่ผู้เรียก

- เมื่อการทำงานพบคำสั่ง **return** ก็จะคืนการทำงานกลับไปสู่ผู้เรียก method
- สำหรับ method ที่ไม่มีการคืนผลลัพธ์
 - หลังคำว่า **return** ปิดท้ายด้วย ; เลย
 - ถ้าคำสั่งท้ายสุดของ method ไม่ใช้ **return** ก็จะสมมุติว่า **return** หลังบรรทัดท้ายสุด

```
public static void printError(String msg) {  
    if ( msg == null ) return;  
    System.err.println( msg );  
}
```

ตัวเมธ็อด : การคืนการทำงานสู่ผู้เรียก

- สำหรับ method ที่มีการคืนผลลัพธ์
 - หลังคำว่า **return** ต้องตามด้วยนิพจน์
 - ค่าของนิพจน์ที่คืนเป็นผลลัพธ์ ต้องเป็นประเภทเดียวกับที่ประกาศไว้ที่หัว method หรือที่ระบบสามารถจะทำ promotion ให้ได้
 - คืนผลได้แค่ตัวเดียว

```
public static long clip(long a) {  
    if ( a > 0 )  
        return a;  
    else  
        return 0; // 0 is promoted to 0L  
}
```

ตัวเมท็อด : Local Variables

- Local variables คือตัวแปรที่ใช้ภายในเมท็อด
- ประกาศใน block ใด ใช้ได้เฉพาะใน block นั้น หรือ block ที่ซ้อนอยู่ชั้นในๆ ของ block นั้น
- ประกาศที่จุดใด ใช้ได้หลังจุดนั้นภายใน block

```
public static void test(int a) {  
    int z = 7;  
    { int x = z; System.out.println( x ); }  
    System.out.println( x ); // wrong  
    {  
        System.out.println( x ); // wrong  
        int x = z; System.out.println( x );  
        { int y = z; System.out.println( y ); }  
        { int x = z; System.out.println( x ); } // wrong  
    }  
}
```

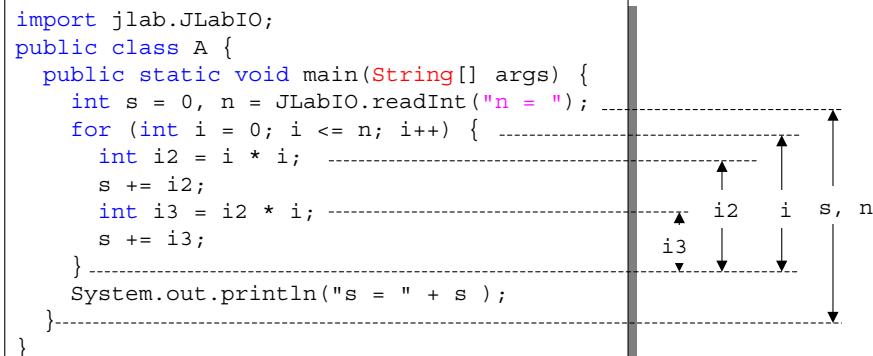
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 21

ตัวเมท็อด : Local Variables

- ถูกสร้างใหม่ทุกครั้งที่ทำงานถึงจุดที่ประกาศ (ระบบสร้างให้โดยไม่ได้เติมค่าเริ่มต้น)
- ถูกทำลายหมดหลังจบ block ที่ถูกประกาศ
- ตัวแปรต่างๆ ในพารามิเตอร์ก็เป็น local variables

```
import jlab.JLabIO;  
public class A {  
    public static void main(String[] args) {  
        int s = 0, n = JLabIO.readInt("n = ");  
        for (int i = 0; i <= n; i++) {  
            int i2 = i * i; -----  
            s += i2;  
            int i3 = i2 * i; -----  
            s += i3;  
        }  
        System.out.println("s = " + s);  
    }  
}
```



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 22

สิงที่มักผิดบ่อย

```
public static printError(String msg) {  
    System.err.println( msg );  
}  
  
public static int abs(int a) {  
    if ( a < 0 ) a = -a;  
}  
  
public static float max(float x, y) {  
    if ( x > y ) return x; else return y;  
}  
  
public static double length(double dx, double dy) {  
    double dx = Math.abs(dx);  
    return Math.sqrt( dx*dx + dy*dy );  
}  
  
public static float getBlackColor() {  
    return 0.0;  
}
```

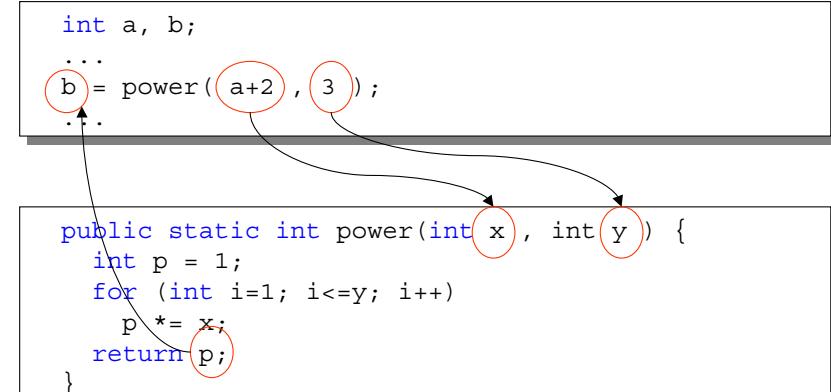
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 23

การเรียกใช้เมท็อด

- การส่งข้อมูลไป (arguments)
- การรับผลที่ได้คืนมา (returned value)
- ไม่มีข้อมูลส่งไป หรือไม่มีผลคืนมา ก็ได้

```
int a, b;  
...  
b = power( a+2, 3 );  
...  
  
public static int power(int x, int y) {  
    int p = 1;  
    for (int i=1; i<=y; i++)  
        p *= x;  
    return p;  
}
```



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 24

การเรียกใช้เมท็อด : การส่งข้อมูล

- จำนวนข้อมูลที่ส่งต้องตรงกับที่ประกาศไว้ที่หัวเมท็อด
- สิ่งที่ส่งไปต้องเป็นประเภทที่ตัวแปรที่หัวเมท็อดรับได้ (ถ้าจำเป็น ระบบจะทำ argument promotion ให้)

```
public static float max(float x, float y) {  
    . . .  
}
```

ถูก

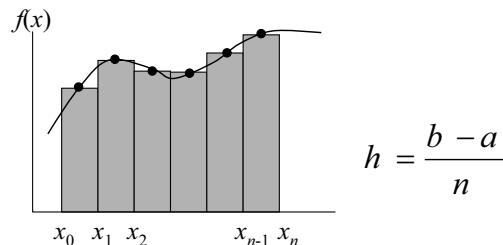
```
x = max( (float) (a/3.0), 1f );  
y = max( Float.parseFloat(str), 1 ); // 1 -> 1.0f
```

ผิด

```
x = max( (float) (a/2) ); // wrong number of args  
y = max( "1.0", 1 ); // can't promote "1.0" to 1f
```

ตัวอย่าง : Integration

- แบ่งพื้นที่ใต้กราฟเป็นแบบๆ
- หาพื้นที่ของแต่ละแบบ และหาผลรวม
- ใช้ midpoint rule ประมาณพื้นที่ใต้กราฟดังนี้



$$\int_a^b f(x) dx \approx \sum_{i=1}^n h f\left(\frac{x_{i-1} + x_i}{2}\right)$$

การเรียกใช้เมท็อด : ผลที่คืนกลับ

- ในการนี้เมท็อดมีการคืนค่า ก็ต้องมีตัวแปรซึ่งมีประเภทที่รับค่าของข้อมูลที่คืนมาได้

```
public static float max(float x, float y) {  
    . . .  
}
```

ถูก

```
double x = max( 1.0f, (int) (x/3) );  
int y = (int) max( 1, 2 );  
float z = max( max(x, y), -10.5f ) ;
```

ผิด

```
long x = max( 4.0f, 2.3f ); // float to long  
float y = max( 4.0, 1.0 ); // double to float  
max( 2.0f, 1.0f ); // max returns float
```

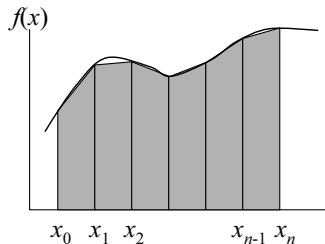
Integration using Midpoint Rule

```
public class Integration {  
    public static void main(String[] args) {  
        System.out.println(simpson(0, 10, 1000));  
    }  
    public static double midpoint(double a, double b, int n) {  
        double h, s1, x;  
        h = (b - a) / n;  
        s1 = 0.0;  
        x = a + h / 2.0;  
        for (int i = 1; i <= n; i++) {  
            s1 += h*f(x);  
            x += h;  
        }  
        return s1;  
    }  
    public static double f(double x) {  
        return 2 * x * x;  
    }  
}
```

$$\int_a^b f(x) dx \approx \sum_{i=1}^n h f\left(\frac{x_{i-1} + x_i}{2}\right)$$

ตัวอย่าง : Integration

- แบ่งพื้นที่ใต้กราฟเป็นแบบๆ
- หาพื้นที่ของแต่ละแบบ แล้วหาผลรวม
- ใช้ Simpson's rule ประมาณพื้นที่ใต้กราฟดังนี้



$$h = \frac{b - a}{n}$$

$$\int_a^b f(x) dx \approx \frac{h}{3} \left(f(a) + 4 \left(\sum_{i=1,3,5,\dots}^{n-1} f(a + ih) \right) + 2 \left(\sum_{i=2,4,6,\dots}^{n-2} f(a + ih) \right) + f(b) \right)$$

Integration using Simpson's Rule

```
public class Integration {
    public static void main(String[] args) {
        System.out.println(simpson(0, 10, 1000));
    }
    public static double simpson(double a, double b, int n) {
        double h, s1, s2, x;
        h = (b - a) / n;
        s1 = s2 = 0.0;
        x = a;
        for (int i = 1; i <= n; i++) {
            x += h;
            if (i % 2 == 1) // is odd ?
                s1 += f(x);
            else
                s2 += f(x);
        }
        return (h / 3 * (f(a) + 4 * s1 + 2 * s2 + f(b)));
    }
    public static double f(double x) {
        return 2 * x * x;
    }
}
```

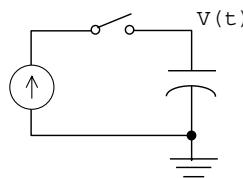
ตัวอย่าง : ความต่างศักย์ของตัวเก็บประจุ

- $t=0$: ตัวเก็บประจุมีประจุเป็นศูนย์
- $t=1$: ปิดสวิตช์ แหล่งจ่ายกระแส จ่าย

$$I(t) = 4(1 - e^{-0.5})e^{-0.5(t-1)}(1 - e^{-t})$$

- ประจุเพิ่ม

$$Q(t) = \int I(t) dt$$



- ความต่างศักย์เพิ่ม

$$V(t) = \frac{Q(t)}{C}$$

Voltage Across a Capacitor

```
public class CapacitorVoltage {
    public static void main(String[] args) {
        double c = 0.05; // capacitance
        double v, q; // voltage and charge
        double a = 1.0, b = 10.0; // 1 to 10 seconds

        for (double t = a; t <= b; t++) {
            q = simpson(a, t, 1000);
            v = q / c;
            System.out.println(t + "\t" + q + "\t" + v);
        }
    }
    public static double simpson(double a, double b, int n) {
        ...
    }
    public static double f(double x) {
        return 4 * (1 - Math.exp(-0.5))
               * Math.exp(-0.5 * (t - 1))
               * (1 - Math.exp(-t));
    }
}
```

Method Overloading

- ชื่อเมท็อดซ้ำได้ ถ้ารายการพารามิเตอร์ไม่เหมือนกัน
- ระบบจะเรียกเมท็อดที่เหมาะสมให้

```
public static int max(int x, int y) {  
    if ( x > y ) return x; else return y;  
}  
public static double max(double x, double y) {  
    if ( x > y ) return x; else return y;  
}  
public static double max(double x, double y, double z) {  
    if ( x > y && x > z ) return x;  
    else if ( y > x && y > z ) return y;  
    else return z;  
}
```

Method Overloading

- ไม่สามารถ overload เมท็อดที่รายการพารามิเตอร์ เหมือน แต่ประเภทข้อมูลที่คืนไม่เหมือน
- ความเหมือนไม่เหมือนของพารามิเตอร์ อยู่ที่ประเภท ข้อมูล ไม่เกี่ยวกับชื่อที่ใช้

```
public static int max(int x, int y) {  
    if ( x > y ) return x; else return y;  
}  
  
public static double max(int x, int y) {  
    if ( x > y ) return (double) x; else return (double) y;  
}  
  
public static int max(int a, int b) {  
    if ( a > b ) return a; else return b;  
}
```

2110101 : การทำโปรแกรมคอมพิวเตอร์

Arrays

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อ

- อาเรย์
 - การประกาศ การสร้าง การใช้
- การส่งอาเรย์ให้เมธ็อด
- การค้นข้อมูล
- การเรียงลำดับข้อมูล
- อาเรย์หลายมิติ

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 2

โปรแกรมเก็บสถิติคะแนนสอบ

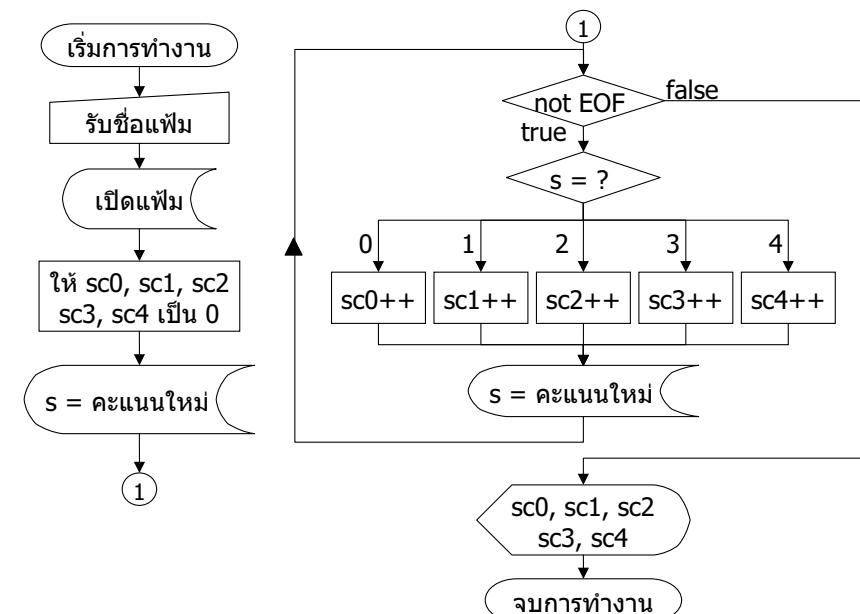
- มีแฟ้มข้อมูลเก็บรหัสนักเรียน + คะแนน
- คะแนนมีตั้งแต่ 0 ถึง 4 (จำนวนเต็ม)
- อยากรู้ว่ามีนักเรียนได้ 0 กี่คน 1 กี่คน ... 4 กี่คน

4531001321	3
4531003621	3
4531004221	4
4531005921	3
4531006521	3
4531101421	4
4531102021	4
4531102221	0



คะแนน	จำนวนนักเรียน
0	1
1	0
2	0
3	4
4	3

ผังงานการเก็บสถิติคะแนนสอบ



โปรแกรมเก็บสถิติคะแนนสอบ

```
import jlab.JLabIO;
import java.io.*;
public class Frequency1 {
    public static void main(String[] args) throws IOException {
        String txt, t;
        int sc, score0, score1, score2, score3, score4;
        BufferedReader in = JLabIO.openFile(JLabIO.readString(">"));
        score0 = score1 = score2 = score3 = score4 = 0;
        while ((txt = in.readLine()) != null) {
            t = txt.substring(10, txt.length());
            switch (Integer.parseInt(t.trim())) {
                case 0 : score0++; break;
                case 1 : score1++; break;
                case 2 : score2++; break;
                case 3 : score3++; break;
                case 4 : score4++; break;
                default : System.out.println("data error -> " + txt);
            }
        }
        in.close();
        // Five more lines to display score frequencies
    }
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 5

ใช้อาร์เรย์ดีกว่า

1	0	0	4	3
score0	score1	score2	score3	score4

ตัวแปร 5 ตัว 5 ชื่อ

0	1	2	3	4	
score	1	0	0	4	3
	score[0]	score[1]	score[2]	score[3]	score[4]

นาทีที่เก็บข้อมูลต่างๆ มารวมกลุ่มกัน 1 กลุ่ม 1 ชื่อ
โดยแต่ละที่เก็บมีหมายเลขกำกับ

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 6

โปรแกรมเก็บสถิติคะแนนสอบ (ใช้อาร์เรย์)

```
import jlab.JLabIO;
import java.io.*;
public class Frequency2 {
    public static void main(String[] args) throws IOException {
        String txt, t;
        int sc;
        int [] score = new int[5]; // declare + allocate array
        BufferedReader in = JLabIO.openFile(JLabIO.readString(">"));

        for (int i=0; i<5; i++) score[i] = 0; // unnecessary

        while ((txt = in.readLine()) != null) {
            t = txt.substring(10, txt.length());
            sc = Integer.parseInt(t.trim());
            score[sc]++;
        }

        for (int i=0; i<5; i++)
            System.out.println( i + " -> " + score[i] );
    }
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 7

แຄוจำดับ (Arrays)

- อาเรย์คือที่เก็บรายการของข้อมูลประเภทเดียวกัน
- อาเรย์เก็บข้อมูลทั้งหลายติดๆ กันไปในหน่วยความจำ เสมือนนำข้อมูลมาเก็บใน "ช่อง" ที่วางเรียงกัน
- แต่ละอาเรย์มีชื่อกำกับ
- ใช้หมายเลขช่องระบุตำแหน่งของข้อมูลในอาเรย์

0	1	2	3	4	5	6	7	8	9	10	11	
a	23	2	3	14	5	99	9	8	7	44	23	3

System.out.println(a[6] + a[9]);

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 8

การประกาศและการสร้างอาร์เรย์

- ก่อนใช้อาร์เรย์ ต้อง
 - ประกาศตัวแปรอาร์เรย์
 - สร้างตัวอาร์เรย์

```
int [] data;
```

ประกาศตัวแปรชื่อว่า data มีไว้อ้างอิงอาร์เรย์ของ int

```
data = new int[10];
```

สร้างอาร์เรย์ของ int จำนวน 10 ช่อง และให้ data เป็นตัวแปรสำหรับอ้างอิงอาร์เรย์ที่ได้สร้างขึ้น

```
int [] data = new int[10];
```

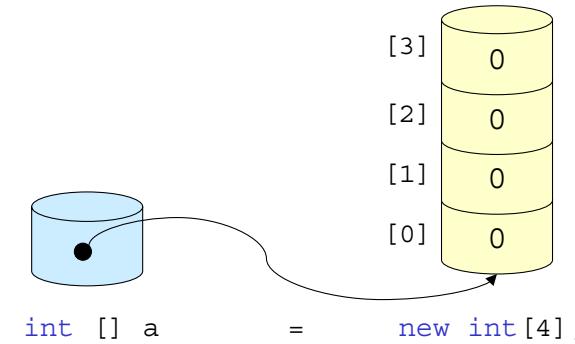
ประกาศตัวแปรและสร้างอาร์เรย์พร้อมกันเลยก็ได้

ข้อสังเกต : การประกาศ+การสร้าง

- การประกาศตัวแปรอาร์เรย์ ไม่ได้เป็นการสร้างตัวอาร์เรย์
- ขนาดของอาร์เรย์ถูกกำหนดตอนสร้างตัวอาร์เรย์ ไม่ใช่ตอนประกาศตัวแปร
- ขนาดของอาร์เรย์เป็นค่าของนิพจน์ก็ได้
(`data = new int[3*n + 1];`)
- อาร์เรย์ที่สร้างขึ้น ไม่สามารถเพิ่มหรือลดขนาดได้
- ระบบจะตั้งค่าเริ่มต้นของข้อมูลในอาร์เรย์โดยอัตโนมัติ
(ให้ค่าศูนย์กับอาร์เรย์ของจำนวน และ ให้ค่า `false` กับอาร์เรย์ของ `boolean`)

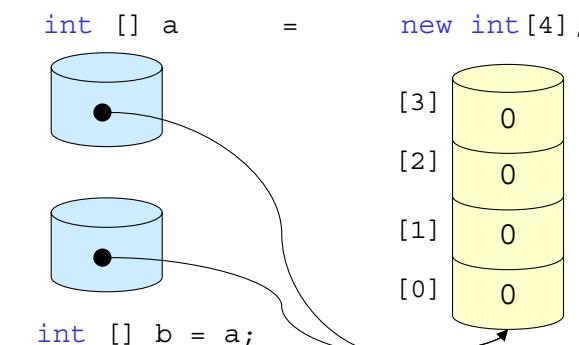
ตัวแปรอาร์เรย์กับตัวอาร์เรย์

- การประกาศตัวแปรอาร์เรย์ เป็นการสร้างที่เก็บ "ที่อยู่"
(reference) ของอาร์เรย์ที่จะสร้าง



ข้อสังเกต : การอ้างอิงอาร์เรย์

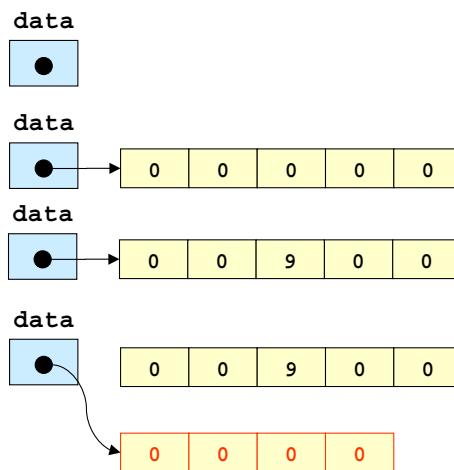
- อาร์เรย์เดวนี้ อาจมีตัวแปรหลายตัวอ้างอิงได้



a กับ b มีค่าเท่ากัน หมายความว่า อ้างอิงอาร์เรย์เดียวกัน ตั้งนั้น a[i] กับ b[i] อ้างอิงที่เก็บข้อมูลที่เดียวกัน

สร้างใหม่ได้ แต่ของเดิมหาย

```
int[] data;  
  
data = new int[4];  
  
data[2] = 9;  
  
data = new int[3];
```



ประกาศ+สร้าง+ตั้งค่าเริ่มต้น

- ใช้ initializer list ระบุค่าเริ่มต้น
- ไม่ต้องกำหนดจำนวนช่องที่จะสร้าง
- จำนวนช่องที่สร้างเท่ากับจำนวนข้อมูลใน list

```
int[] data = new int[4];  
data[0] = 23;  
data[1] = 3;  
data[2] = 21;  
data[3] = 47;
```

```
int[] data = {23, 3, 21, 47};
```

```
int[] data;  
data = {23, 3, 21, 47};
```

เขียนแยกเป็นสองบรรทัดแบบนี้ไม่ได้

การใช้ข้อมูลในอาร์เรย์

- ใช้หมายเลขช่อง (เรียกว่า index) ของอาร์เรย์เป็นตัวอ้างอิงข้อมูลในอาร์เรย์ เช่น data[2] เป็นการอ้างอิงข้อมูลช่องที่ index 2 ในอาร์เรย์ data
- index ของอาร์เรย์เริ่มที่ 0 เสมอ
- ถ้าอาร์เรย์มีขนาด n ตัว index ต้องเป็นจำนวนเต็ม มีค่าได้ตั้งแต่ 0 ถึง n-1

```
public class Array1 {  
    public static void main(String [] args) {  
        int [] data = new int[100];  
        int sum = 0;  
        for(int i = 0; i < 100; i++)  
            sum += data[i];  
        System.out.println( sum );  
    }  
}
```

i = 0; i < 100

การอ้างอิงข้อมูลแบบผิดๆ

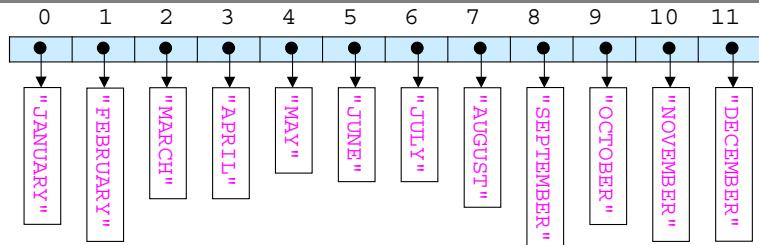
- ถ้าอาร์เรย์มีขนาด n ตัว index ต้องเป็นจำนวนเต็มมีค่าได้ตั้งแต่ 0 ถึง n-1
- index ไม่ใช้จำนวนเต็ม - compile error
- index เป็นจำนวนเต็มที่อยู่นอกช่วง
 - เกิด **ArrayIndexOutOfBoundsException** ขณะทำงาน

```
public class Array3 {  
    public static void main(String [] args) {  
        int [] data = new int[100];  
        System.out.println( data[100] );  
    }  
}
```

```
JLab>java Array3  
java.lang.ArrayIndexOutOfBoundsException  
at Array3.main(Array3.java:4)  
Exception in thread "main"  
JLab>
```

โปรแกรมแสดงชื่อเดือน

```
import jlab.JLabIO;
public class Month {
    public static void main(String [] args) {
        String month[] =
            {"JANUARY", "FEBRUARY", "MARCH", "APRIL",
             "MAY", "JUNE", "JULY", "AUGUST", "SEPTEMBER",
             "OCTOBER", "NOVEMBER", "DECEMBER"};
        int m = JLabIO.readInt("Enter month number : ");
        if (1<=m && m<=12) System.out.println(month[m-1]);
    }
}
```

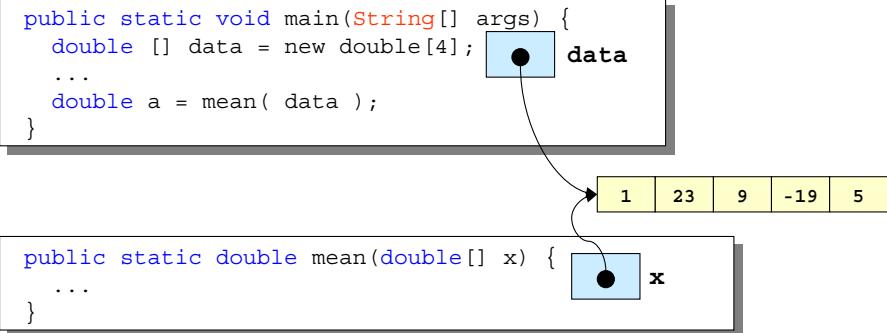


2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 17

การส่งอาร์เรย์ไปยังเมธ็อดอื่น

- ใช้ชื่ออาร์เรย์แทนทั้งอาร์เรย์ (ไม่มี [])
- การส่งอาร์เรย์จะไม่ได้ส่งข้อมูลของทั้งอาร์เรย์ไป แต่จะส่งตำแหน่ง (reference) ของอาร์เรย์ไปแทน



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 18

ถ้าอยากรู้ขนาดของอาร์เรย์

- เติม .length ตามหลังชื่ออาร์เรย์

```
public class Array2 {
    public static void main(String [] args) {
        int [] data = new int[100];
        System.out.println( sum(data) );
    }

    public static int sum(int [] d) {
        int sum = 0;
        for(int i = 0; i < d.length; i++)
            sum += d[i];
        return sum;
    }
}
```

อย่าจำสับสนกับความยาว String ซึ่งใช้ .length() เพราะเป็นเมธ็อด

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 19

โปรแกรมนับความถี่ตัวอักษรต่างๆ

```
import jlab.JLabIO;
import java.io.*;
public class CountChars {
    public static void main(String[] args) throws IOException {
        String txt;
        int[] freq = new int[26];
        BufferedReader in = JLabIO.openFile(JLabIO.readString(">"));
        while ((txt = in.readLine()) != null) {
            int txtLen = txt.length();
            for (int i = 0; i < txtLen; i++) {
                char c = Character.toUpperCase(txt.charAt(i));
                if ('A' <= c && c <= 'Z') freq[c - 'A']++;
            }
            for (int i = 0; i < freq.length; i++) {
                if (freq[i] > 0)
                    System.out.println((char)('A' + i) + " --> " + freq[i]);
            }
            in.close();
        }
    }
}
```

ใน Java char เป็นจำนวนเต็มแบบ
ไม่ติดลบด้วย (unsigned integer)

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 20

เมธอดหาค่าเฉลี่ย ค่าเบี่ยงเบน

```

public static double mean(double[] x) {
    double sum = 0.0, m = 0.0;
    if (x.length > 0) {
        for (int i = 0; i < x.length; i++)
            sum += x[i];
        m = sum / x.length;
    }
    return m;
}

public static double stddev(double[] x) {
    double dev = 0;
    double sumsq, m = mean(x);
    if (x.length > 1) {
        for (int i = 0; i < x.length; i++)
            sumsq += (x[i] - m) * (x[i] - m);
        dev = sum / (x.length - 1);
    }
    return dev;
}

```

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 21

การค้นข้อมูลแบบลำดับ

Requirement

- เมธอดเพื่อการค้นข้อมูลในอาร์เรย์

Analysis

- รับ data เป็นอาร์เรย์ของจำนวนเต็ม
- รับ key เป็นจำนวนเต็มที่ต้องการค้นในอาร์เรย์
- คืน index ตัวหนึ่งของ data ที่ข้อมูลมีค่าเท่ากับ key แต่ถ้าไม่ปรากฏ key ใน data ให้คืนค่า -1
- ใช้วิธีการค้นแบบลำดับ (sequential search)

```
public static int seqSearch( int [] data, int key )
```

0	1	2	3	4	5	6
data	23	2	3	14	5	99

key

เรียก seqSearch(data, key) จะได้ 4

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 22

การค้นข้อมูลแบบลำดับ

```
public static int sequentialSearch( int [] data, int key )
```

0	1	2	3	4	5	6
data	23	2	3	14	5	99

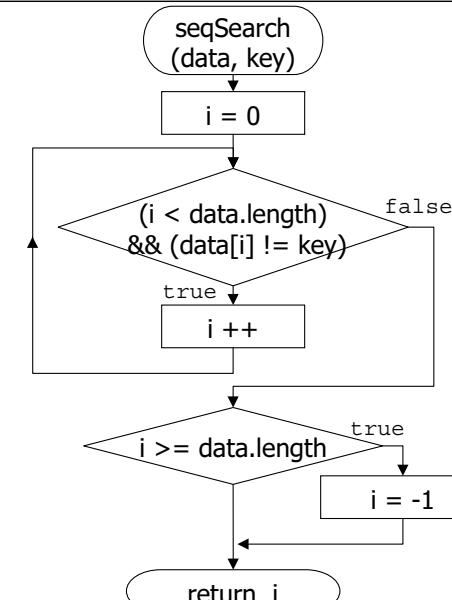
key 5		
i	data[i]	data[i] == key
0	23	false
1	2	false
2	3	false
3	14	false
4	5	true
return i		

key 8		
i	data[i]	data[i] == key
0	23	false
1	2	false
2	3	false
3	14	false
4	5	false
5	99	false
6	9	false
7		
return -1		

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 23

ผังงานการค้นข้อมูลแบบลำดับ



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 24

เมธีดการค้นข้อมูลแบบลำดับ

```
public class TestSeqSearch {  
  
    public static int seqSearch(int[] data, int key) {  
        int i = 0;  
        while (i < data.length && data[i] != key) {  
            i++;  
        }  
        if (i >= data.length) i = -1;  
        return i;  
    }  
  
    // main is used for testing  
    public static void main(String[] args) {  
        int[] data = {23, 2, 3, 14, 5, 99, 9};  
        System.out.println(seqSearch(data, 5));  
        System.out.println(seqSearch(data, 8));  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 25

การค้นข้อมูลแบบลำดับ (ถอยหลัง)

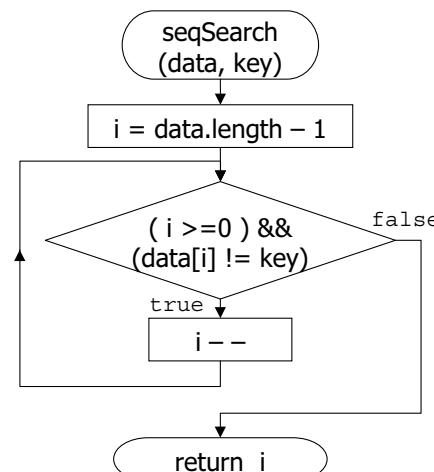
```
public static int sequentialSearch( int [] data, int key )
```

0	1	2	3	4	5	6
data	23	2	3	14	5	99

key 5		
i	data[i]	data[i] == key
6	9	false
5	99	false
4	5	true
return i		

key 8		
i	data[i]	data[i] == key
6	9	false
5	99	false
4	8	false
3	14	false
2	3	false
1	2	false
0	23	false
-1		
return i		

ผังงานการค้นข้อมูลแบบลำดับ (ถอยหลัง)



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 27

เมธีดการค้นข้อมูลแบบลำดับ (ถอยหลัง)

```
public class TestSeqSearch {  
  
    public static int seqSearch(int[] data, int key) {  
        int i = data.length - 1;  
        while (i >= 0 && data[i] != key) {  
            i--;  
        }  
        return i;  
    }  
  
    // main is used for testing  
    public static void main(String[] args) {  
        int[] data = {23, 2, 3, 14, 5, 99, 9};  
        System.out.println(seqSearch(data, 5));  
        System.out.println(seqSearch(data, 8));  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

15/7/2002 28

เขียนได้หลายแบบ

```
public static int seqSearch(int[] data, int key) {  
    int i = data.length;  
    while (i >= 0 && data[i] != key) i--;  
    return i;  
}
```

```
public static int seqSearch(int[] data, int key) {  
    int i;  
    for (i = data.length - 1;  
         i >= 0 && data[i] != key;  
         i--) ;  
    return i;  
}
```

อย่าลูกเล่นมาก

```
public static int seqSearch(int[] data, int key) {  
    int i = data.length - 1;  
    while (data[i] != key && i >= 0) i--;  
    return i;  
}
```

ตรวจสอบ $i \geq 0$ หลังการตรวจสอบ $\text{data}[i] \neq \text{key}$

```
public static int seqSearch(int[] data, int key) {  
    int i = data.length - 1;  
    while (i-- >= 0 && data[i] != key) ;  
    return i;  
}
```

i ถูกลดค่าก่อนที่จะใช้ $\text{data}[i]$

```
public static int seqSearch(int[] data, int key) {  
    int i = data.length - 1;  
    while (i >= 0 && data[i--] != key) ;  
    return i;  
}
```

i ถูกลดค่าทุกครั้งหลังเปรียบเทียบ
 $\text{return } i$ จะให้ผลผิด

การหาตำแหน่งที่มีค่ามากสุด

• Requirement

- เมธอดเพื่อการหาตำแหน่งที่มีค่ามากสุดในอารเรย์

• Analysis

- รับ data เป็นอารเรย์ของจำนวนเต็ม
- คืน index ของอารเรย์ที่มีค่าไม่น้อยกว่าค่าใดๆ ในอารเรย์
- ใช้วิธีการหาค่ามากสุดแบบลำดับ

```
public static int max( int [] data )
```

0	1	2	3	4	5	6
23	2	3	14	5	99	9

เรียก $\text{max}(\text{data})$ จะได้ 5

เมธอดการหาตำแหน่งที่มีค่ามากสุด

```
public class TestMax {  
    public static int max(int [] data) {  
        return max(data, data.length);  
    }  
    // find the max among the first k elements in the array  
    public static int max(int [] data, int k) {  
        int m = -1;  
        if (k <= data.length && k > 1) {  
            m = 0;  
            for (int i = 1; i < k; i++)  
                if (data[m] < data[i]) m = i;  
        }  
        return m;  
    }  
    public static void main(String[] args) {  
        int[] data = {23, 2, 3, 14, 5, 99, 9};  
        System.out.println(max(data));  
        System.out.println(max(new int[0])); // zero size array  
    }  
}
```

การเรียงลำดับข้อมูล

• Requirement

- เมท็อดเพื่อการเรียงลำดับข้อมูลในอาร์เรย์

• Analysis

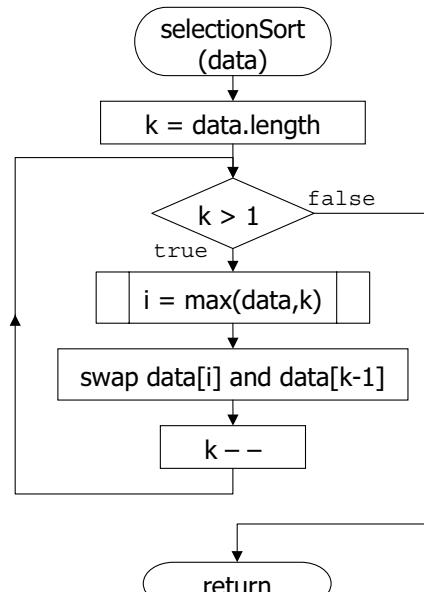
- รับ data เป็นอาร์เรย์ของจำนวนเต็ม
- จัดลำดับข้อมูลใน data ให้เรียงลำดับจากน้อยไปมาก
 $\text{data}[0] \leq \text{data}[1] \leq \text{data}[2] \leq \dots$
- ใช้วิธีการเรียงลำดับแบบเลือก (selection sort)

```
public static void selectionSort( int [] data )
```

0	1	2	3	4	5	6
data	23	2	3	14	5	99
data	2	3	5	9	14	23

ก่อน
หลัง

ผังงานการเรียงลำดับแบบเลือก



การเรียงลำดับแบบเลือก

• มองอาร์เรย์เป็นสองส่วน

- ส่วนขวา เรียงเสร็จแล้ว
- ส่วนซ้าย ยังไม่เรียง

23	2	3	14	5	99	9
23	2	3	14	5	9	99
23	2	3	14	5	9	99
9	2	3	14	5	23	99
9	2	3	14	5	23	99
9	2	3	5	14	23	99
9	2	3	5	14	23	99
5	2	3	9	14	23	99
5	2	3	9	14	23	99
3	2	5	9	14	23	99
3	2	5	9	14	23	99
2	3	5	9	14	23	99

เมท็อดการเรียงลำดับแบบเลือก

```
public class TestSort {  
    public static void selectionSort(int[] data) {  
        int i, t;  
        for (int k = data.length; k > 1; k--) {  
            i = max(data, k);  
            t = data[i];  
            data[i] = data[k - 1];  
            data[k - 1] = t;  
        }  
    }  
    public static void main(String[] args) {  
        int[] data = {23, 2, 3, 14, 5, 99, 9};  
        selectionSort(data);  
        printArray(data);  
    }  
    public static void printArray(int[] data) {  
        for (int i = 0; i < data.length; i++)  
            System.out.print(data[i] + " ");  
        System.out.println();  
    }  
    public static int max(int[] data, int k) {  
        ...  
    }  
}
```

Arrays.binarySearch ของ Java API

- ค้นข้อมูล (ข้อมูลในอาร์เรย์ต้องเรียงลำดับแล้ว)

```
int binarySearch(byte[] a, byte key)
int binarySearch(short[] a, short key)
int binarySearch(int[] a, int key)
int binarySearch(long[] a, long key)
int binarySearch(float[] a, float key)
int binarySearch(double[] a, double key)
int binarySearch(char[] a, char key)
```

```
import java.util.Arrays;

public class TestSort {
    public static void main(String[] args) {
        int[] data = {2, 3, 5, 9, 14, 23, 99};
        System.out.println(Arrays.binarySearch(data, 5));
        System.out.println(Arrays.binarySearch(data, 8));
    }
}
```

Arrays.equals ของ Java API

- ตรวจว่าอาร์เรย์สองแกรมีค่าเท่ากันหรือไม่

```
boolean equals(boolean[] a, boolean[] b)
boolean equals(byte[] a, byte[] b)
boolean equals(short[] a, short[] b)
boolean equals(int[] a, int[] b)
boolean equals(long[] a, long[] b)
boolean equals(float[] a, float[] b)
boolean equals(double[] a, double[] b)
boolean equals(char[] a, char[] b)
```

Arrays.fill ของ Java API

- เติมค่าเริ่มต้นให้ช่องต่างๆ เท่ากับค่าที่กำหนดให้

```
void fill(boolean[] a, boolean val)
void fill(byte[] a, byte val)
void fill(short[] a, short val)
void fill(int[] a, int val)
void fill(long[] a, long val)
void fill(float[] a, float val)
void fill(double[] a, double val)
void fill(char[] a, char val)
```

```
void fill(boolean[] a, int from, int to, boolean val)
void fill(byte[] a, int from, int to, byte val)
void fill(short[] a, int from, int to, short val)
void fill(int[] a, int from, int to, int val)
void fill(long[] a, int from, int to, long val)
void fill(float[] a, int from, int to, float val)
void fill(double[] a, int from, int to, double val)
void fill(char[] a, int from, int to, char val)
```

Arrays.sort ของ Java API

- เรียงลำดับข้อมูล

```
void sort(boolean[] a)
void sort(byte[] a)
void sort(short[] a)
void sort(int[] a)
void sort(long[] a)
void sort(float[] a)
void sort(double[] a)
void sort(char[] a)
```

```
void sort(boolean[] a, int from, int to)
void sort(byte[] a, int from, int to)
void sort(short[] a, int from, int to)
void sort(int[] a, int from, int to)
void sort(long[] a, int from, int to)
void sort(float[] a, int from, int to)
void sort(double[] a, int from, int to)
void sort(char[] a, int from, int to)
```

System.arraycopy ของ Java API

- copy ข้อมูลจากอาร์เรย์ไปยังอีกอาร์เรย์หนึ่ง

```
public class ArrayCopy {  
    public static void main(String[] args) {  
        int[] a = {23, 2, 3, 14, 5, 99, 9};  
        int[] b = new int[4];  
        System.arraycopy(a, 2, b, 0, 3);  
    }  
}
```



a	0	1	2	3	4	5	6
	23	2	3	14	5	99	9

b	0	1	2	3
	3	14	5	0

Initializer List

```
int [][] b = { {1, 2, 3, 4, 5},  
               {5, 4, 3, 2, 1},  
               {10, 20, 30, 40, 50} };
```

b	0	1	2	3	4
0	1	2	3	4	5
1	5	4	3	2	1
2	10	20	30	40	50

อาร์เรย์หลายมิติ (Multidimensional Array)

```
int [] a = new int [7];  
a[2] = 3;
```

หนึ่งมิติ
(vector)

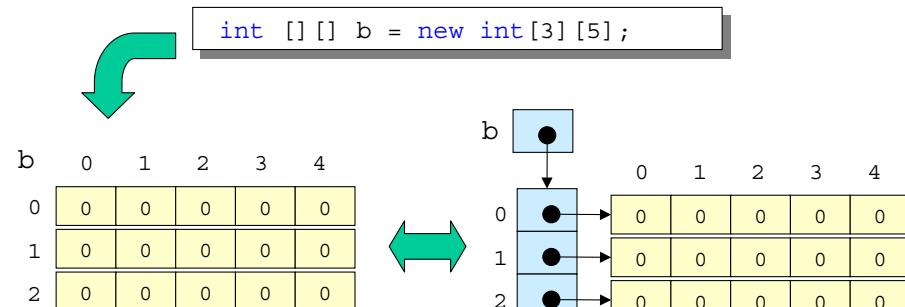
a	0	1	2	3	4	5	6
	0	0	3	0	0	0	0

```
int [] [] b = new int [3] [5];  
b[1] [3] = 4;
```

สองมิติ
(matrix)

b	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	4	0
2	0	0	0	0	0

อาร์เรย์ของอาร์เรย์



```
int [] [] b;  
b = new int [3] [] ;  
b[0] = new int [5] ;  
b[1] = new int [5] ;  
b[2] = new int [5] ;
```



เมท็อดแสดงค่าในอาร์เรย์สองมิติ

```
public class TestPrintArray2 {  
    public static void main(String[] args) {  
        int[][] b = {{1, 2, 3, 4, 5},  
                     {5, 4, 3, 2, 1},  
                     {10, 20, 30, 40, 50}};  
        printArray2(b);  
    }  
  
    public static void printArray2(int[][] a) {  
        for (int i = 0; i < a.length; i++) {  
            for (int j = 0; j < a[i].length; j++)  
                System.out.print(a[i][j] + "\t");  
            System.out.println();  
        }  
    }  
}
```

```
JLab>java TestPrintArray2  
1      2      3      4      5  
5      4      3      2      1  
10     20     30     40     50  
JLab>
```

การบวกเมทริกซ์

- Requirement
 - เมท็อดสำหรับการบวกเมทริกซ์
- Analysis
 - รับ a และ b เป็นอาร์เรย์สองมิติของ int
 - คืนอาร์เรย์สองมิติซึ่งเป็นผลบวกของ a กับ b

$$c_{ij} = a_{ij} + b_{ij}$$

เมท็อดสำหรับการบวกเมทริกซ์

```
public class TestAddMatrices {  
    public static void main(String[] args) {  
        int[][] b = {{1, 2, 3, 4, 5},  
                     {5, 4, 3, 2, 1},  
                     {10, 20, 30, 40, 50}};  
        printArray2(addMatrices(b, b));  
    }  
  
    public static int[][] addMatrices(int[][] a,  
                                      int[][] b) {  
        int[][] c = new int[a.length][a[0].length];  
        for (int i = 0; i < a.length; i++)  
            for (int j = 0; j < a[i].length; j++)  
                c[i][j] = a[i][j] + b[i][j];  
        return c;  
    }  
  
    public static void printArray2(int[][] a) {  
        ...  
    }  
}
```

การคูณเมทริกซ์

- Requirement
 - เมท็อดสำหรับการคูณเมทริกซ์
- Analysis
 - รับ a และ b เป็นอาร์เรย์สองมิติของ int
 - คืนอาร์เรย์สองมิติซึ่งเป็นผลคูณของ a กับ b
 - ถ้า a มีขนาด $[p \times q]$ และ b มีขนาด $[q \times r]$ ผลคูณของ a \times b จะเป็นเมทริกซ์ขนาด $[p \times r]$

$$c_{ij} = \sum_{k=1}^q a_{ik} b_{kj}$$

เมธอดสำหรับการคูณเมตริกซ์

```
public static int[][] mulMatrices(int[][] a,
                                  int[][] b) {
    int[][] c = new int[a.length][b[0].length];
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < b[i].length; j++) {
            c[i][j] = 0;
            for (int k = 0; k < b.length; k++) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    return c;
}
```

$$c_{ij} = \sum_{k=1}^q a_{ik} b_{kj}$$

จำนวนฟีโบนัคชี

```
import jlab.JLabIO;
public class TestMulMatrices {
    public static void main(String[] args) {
        int[][] f1 = {{0, 1}, {1, 1}};
        int[][] fn = {{0, 1}, {1, 1}};
        int n = JLabIO.readInt("Fibonacci n = ");
        for (int i = 1; i < n; i++)
            fn = mulMatrices(f1, fn);
        System.out.println("f(" + n + ") = "
                           + fn[0][1]);
    }

    public static int[][] mulMatrices(int[][] a,
                                      int[][] b) {
        ...
        
$$\begin{bmatrix} f_{n-1} & f_n \\ f_n & f_{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^n$$

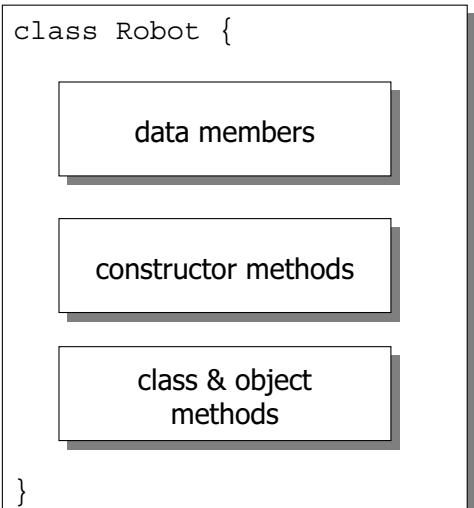
    }
}
```

2110101 : การทำโปรแกรมคอมพิวเตอร์

Classes & Objects

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

องค์ประกอบของคลาส



หัวข้อ

- องค์ประกอบของคลาส
- การประการศ การสร้าง และการทำลายออบเจ็กต์
- Instance variables
- Class methods และ object methods
- Constructor methods
- ตัวอย่าง : คลาส Complex
 - Newton-Raphson
 - Mandelbrot Set

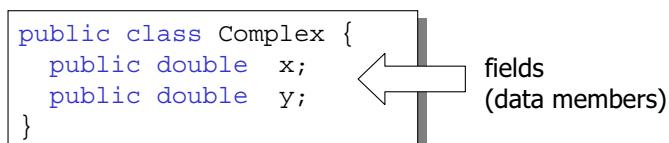
จำนวนเชิงซ้อน

- จำนวนเชิงซ้อน $z = x + iy$
 - x เป็นจำนวนจริง
 - iy เป็นจำนวนจินตภาพ
 - y เป็นจำนวนจริง
 - $i = \sqrt{-1}$ (imaginary unit)
- conjugate $\bar{z} = x - iy$
- absolute square : $|z|^2 = z\bar{z} = (x+iy)(x-iy) = x^2 + y^2$
- ให้ $z_1 = x_1 + iy_1$ และ $z_2 = x_2 + iy_2$
- $z_1 + z_2 = (x_1+x_2) + i(y_1+y_2)$
- ได้เจออีกเยอะใน ไฟฟ้า เครื่องกล โยธา ...

$$e^{i\pi} = -1$$

อย่างไฉลจำนวนเชิงซ้อน

- จำวมมีแต่จำนวนจริง จำนวนเต็ม
- แต่ไม่มีจำนวนเชิงซ้อน ต้องออกแบบเอง
- เขียน class ใหม่ชื่อ Complex
- เขียน class = เขียนนิยามของประเภทข้อมูลแบบใหม่
- เก็บจำนวนเชิงซ้อนหนึ่งจำนวน = เก็บจำนวนจริงสองจำนวน $z = x + iy$

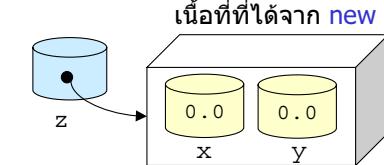


การประกาศและการสร้าง

```
public class Complex {  
    public double x;  
    public double y;  
}
```

นิยามของคลาส

```
Complex z;  
z = new Complex();
```



Complex z;
เป็นการประกาศและสร้างตัวแปร z ซึ่งมีไว้อ้างอิงเนื้อที่เก็บข้อมูล
ที่เป็นแบบ Complex

```
z = new Complex();
```

เป็นการสร้างเนื้อที่เก็บข้อมูลที่เป็นแบบ Complex

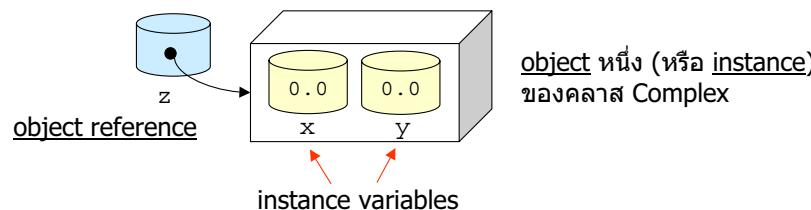
ศัพท์เทคนิค

class ชื่อ Complex

```
public class Complex {  
    public double x;  
    public double y;  
}
```

fields (หรือ data members) ของ class

ตัวแปรอ้างอิง object
ของ class Complex

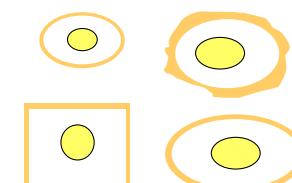


ตำรา กับ ข้าว กับ กับ ข้าว

สูตรใบดาว

- ไข่ไก่นึ่งฟอง
- น้ำมันครึ่งถ้วย

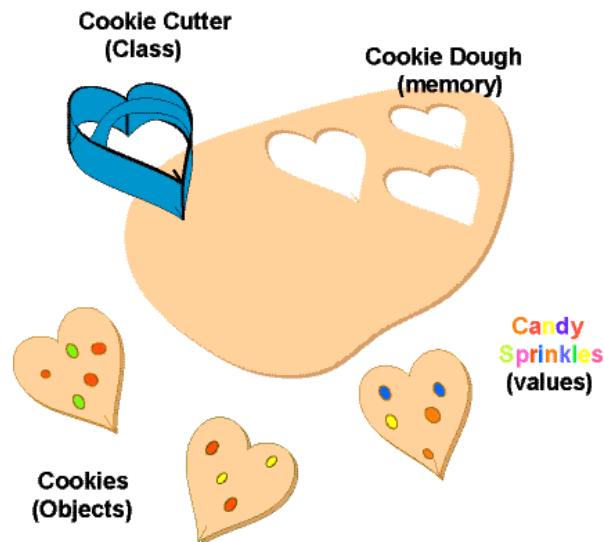
ไข่ดาวที่หยอดแล้วตามสูตร



class บรรยายของคู่ประกอบ

object คือของจริงที่ถูกสร้างขึ้น
object ของ class เดียวกัน แต่
คนละก้อนกัน ก็อาจมีรายละเอียด
ภายในไม่เหมือนกัน

คุกคี



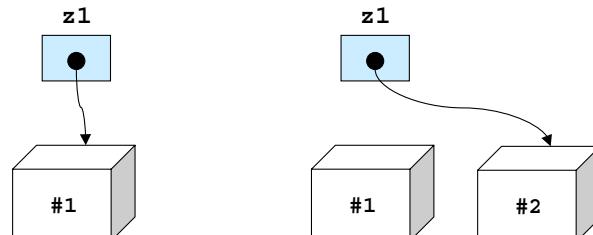
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 9

ขยะ

- Object ที่ถูกสร้างขึ้น แต่ไม่มีตัวแปรได้อ้างอิง
- เนื้อที่สูญเปล่า
- ระบบจาวาจะ "เก็บขยะ" นำเนื้อที่เหล่านี้ไปใช้ใหม่

```
Complex z1 = new Complex(); // #1  
...  
z1 = new Complex(); // #2
```



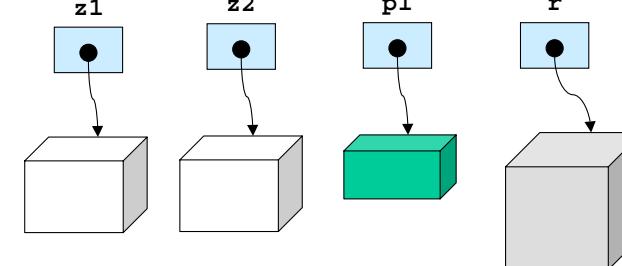
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 11

new เป็นสเมือนโรงงานสร้าง objects

- `new` จะสร้าง object ใหม่ของ class ที่กำหนดให้
- ต้องมีตัวแปรอยู่รับตำแหน่งอ้างอิง object ใหม่

```
Complex z1 = new Complex();  
Complex z2 = new Complex();  
Point p1 = new Point();  
Rectangle r = new Rectangle();
```



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

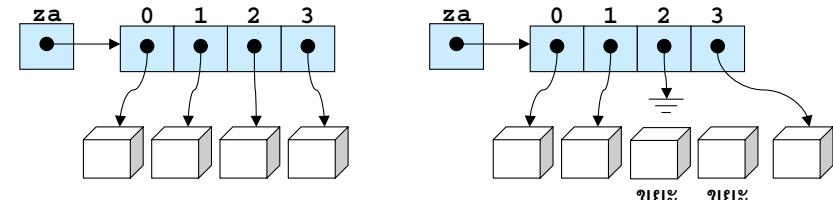
7/8/2002 10

การสร้างและทำลาย object

- สร้างโดยใช้ `new`
- ทำลายโดยอย่าให้มีตัวแปรได้อ้างอิงถึง

```
Complex [] za = new Complex[4];  
for(int i=0; i< za.length; i++) {  
    za[i] = new Complex();  
}  
za[2] = null;  
za[3] = new Complex();
```

`null` เป็นค่าคงตัวพิเศษแทน object reference ที่ไม่ได้อ้างอิงอะไร



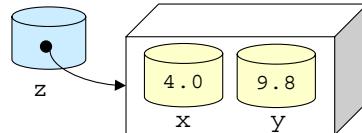
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 12

การใช้ Instance Variables

- instance variables คือตัวแปรต่างๆ ภายในของ object ที่ได้ถูกสร้างขึ้น
- อย่างจะใช้ ให้เขียนในรูป *object.variableName*

```
public class Complex {  
    public double x;  
    public double y;  
}
```



```
Complex z = new Complex();  
  
z.x = 4;  
z.y = 9.8;
```

Methods

- ออกแบบคลาสเพื่อบรรยายลักษณะของคลาส
 - fields บรรยายลักษณะของ object
 - methods บรรยายบริการที่มีให้กระทำกับ object
- คลาส Complex
 - fields :
 - x* และ *y* แทนจำนวนจริงและจินตภาพ
 - methods :
 - หา *absolute*
 - หา *conjugate*
 - หาผลบวกของจำนวนเชิงซ้อนสองจำนวน
 - sort ข้อมูลใน array ของ int ที่กำหนดให้

← ไม่เกี่ยว

การให้ค่าเริ่มต้นกับ instance variable

```
public class Complex {  
    public double x;  
    public double y;  
}
```

ปกติเมื่อ object หนึ่งถูกสร้างขึ้น จะมีการให้ค่า 0 และ false กับ instance variables ต่างๆ ของ object ที่เป็นประเภทจำนวนและ boolean ตามลำดับ

```
public class Complex {  
    public double x = 1.0;  
    public double y = 0.0;  
}
```

ใส่ค่าเริ่มต้นเองก็ได้ เช่นเดียวกับการตั้งค่าเริ่มต้นของตัวแปรภายในเมธ็อด

Class Methods vs. Object Methods

- ออกแบบเมธ็อดให้ผู้อื่นเรียกใช้ได้สองวิธี
 - class methods *className.methodName(...)*

```
double a = Complex.abs(z1);  
Complex z2 = Complex.conjugate(z1);  
Complex z3 = Complex.add(z2, z1);
```

- object methods *objectName.methodName(...)*

```
double a = z1.abs();  
Complex z2 = z1.conjugate();  
Complex z3 = z2.add(z1);
```

ย้อนอดีต

- เคยใช้ class methods มาแล้ว
 - `Math.sqrt(x)`
 - `JLabIO.readInt()`
 - `Array.equals(a, b)`
 - `public static void main(String [] args)`
 - ...
- เคยใช้ object methods มาแล้ว
 - `str.toUpperCase()`
 - `in.readLine()`
 - `System.out.println(msg)`
 - ...

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 17

Class Methods

```
public class Complex {  
    public double x;  
    public double y;  
  
    public static double abs( Complex z ) {  
        return Math.sqrt( z.x * z.x + z.y * z.y );  
    }  
}
```

บรรยายคลาส Complex

```
public class Test {  
    public static void main( String [] args ) {  
        double a = JLabIO.readDouble("real part : ");  
        double b = JLabIO.readDouble("imag part : ");  
        Complex z = new Complex();  
        z.x = a;  
        z.y = b;  
        System.out.println( Complex.abs(z) );  
    }  
}
```

สร้าง object และใช้ method ของ Complex

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 18

Object Methods

```
public class Complex {  
    public double x;  
    public double y;  
  
    public static double abs( ) {  
        return Math.sqrt(this.x*this.x + this.y*this.y);  
    }  
}
```

ตัดคำว่า `static` ทิ้ง ก็
หมายถึง object method

```
public class Test {  
    public static void main( String [] args ) {  
        double a = JLabIO.readDouble("real part : ");  
        double b = JLabIO.readDouble("imag part : ");  
        Complex z = new Complex();  
        z.x = a;  
        z.y = b;  
        System.out.println( z.abs() );  
    }  
}
```

this ใช้อ้างถึง object ที่ถูก
เรียกให้มาทำงานที่เมธอด

Class Methods vs Object Methods

- class method มีคำว่า "static" กำกับที่หัว method
- object method ไม่มีคำว่า "static" กำกับ
- เรียก class methods ใช้ชื่อคลาสนำ
- เรียก object methods ใช้ชื่ออบเจกต์นำ
- คำถาม : บรรทัดใดข้างล่างนี้เรียกใช้ class method
 - `System.getProperties()`
 - `Math.random()`
 - `JLabIO.format(x, 5, 2)`
 - `a.compareTo(b)`
 - `rectangle.setSize(w, h)`
 - `Math.PI`

การตั้งชื่อย่างมี
ระบบจะช่วยให้เข้าใจ
โปรแกรมได้ดีขึ้น

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 19

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 20

ตัวอย่าง Object Methods ของ Complex

```
public class Complex {  
    public double x, y; // data members  
  
    public double abs() {  
        return Math.sqrt(this.x*this.x + this.y*this.y);  
    }  
    public Complex conjugate() {  
        Complex zbar = new Complex();  
        zbar.x = this.x;  
        zbar.y = -(this.y);  
        return zbar;  
    }  
    public Complex add( Complex z ) {  
        Complex sumz = new Complex();  
        sumz.x = this.x + z.x;  
        sumz.y = this.y + z.y;  
        return sumz;  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 21

$$\bar{z} = x - iy$$

$$z_1 + z_2 = (x_1+x_2) + i(y_1+y_2)$$

Object Method ที่คุณมี : toString

- `toString` มีไว้แปลงข้อมูลภายใน object (instance variables) ให้เป็น String ที่คนอ่านเข้าใจ
- โดยทั่วไปใช้เพื่อทำให้แสดงผลได้สะดวก เพราะว่า `System.out.println(z)` จะเรียก `z.toString()` โดยอัตโนมัติ

```
public class Complex {  
    public double x, y;  
    ...  
    public String toString() {  
        return "(" + this.x + ", " + this.y + ")";  
    }  
    ...  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 22

Object Method ที่คุณมี : equals

- ทดสอบ `obj1 == obj2` หมายความว่า `obj1` และ `obj2` เป็น object เดียวกัน หรือไม่
- ทดสอบ `obj1.equals(obj2)` หมายความว่า object `obj1` "เท่ากับ" object `obj2` หรือไม่
- ความหมายของการ "เท่ากัน" ของ object นั้นขึ้น กับผู้ออกแบบ class

```
public class Complex {  
    public double x, y;  
    ...  
    public boolean equals( Complex z ) {  
        return this.x == z.x && this.y == z.y;  
    }  
    ...  
}
```

equals ตามกฎในJAVAจะจุกจิกกว่านี้

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 23

ข้อสังเกต

- เรียกใช้ instance variable `object.variable`
- เรียกใช้ object method `object.method(...)`
- ต่างกันตรงที่มีวงเล็บหรือไม่

```
public class Complex {  
    public double x, y;  
    ...  
    public String toString() {  
        return "(" + this.x + ", " + this.y + ")";  
    }  
    ...  
}
```

```
Complex z1 = new Complex();  
...  
double p = z1.x;  
String s = z1.toString();  
double q = z1.x(); // WRONG  
String t = z1.toString(); // WRONG  
System.out.println( "what am I ?" );
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 24

ย้อนอดีต

- String และ array เป็น object
- String มีเมท็อด length()
- array มี instance variable ชื่อว่า length

```
String s = "Hi Ho Hi Ho;  
int a = new int[100];  
  
System.out.println( s.length() );  
System.out.println( a.length );
```

Constructor Methods

- เป็นเมท็อดพิเศษของคลาส ซึ่งระบบเรียกใช้โดยอัตโนมัติหลัง new คลาสสนับ
- มีไว้เพื่อตั้งค่าเริ่มต้นให้กับ instance variables ของ object ใหม่ซึ่งเพิ่งถูกสร้าง
- เป็นเมท็อดพิเศษ
 - ชื่อต้องเหมือนชื่อ class
 - ต้องไม่ระบุ return type ของเมท็อด

```
public class Complex {  
    public double x, y;  
  
    public Complex( ) {  
        this.x = 0.0;  
        this.y = 0.0;  
    }  
}
```

Constructor Methods

- Overload ได้ด้วยการกำหนดรายการของ parameter ที่แตกต่างกัน

```
public class Complex {  
    public double x, y;  
  
    public Complex( ) {  
        this.x = 0.0;  
        this.y = 0.0;  
    }  
    public Complex( double x1, double y1 ) {  
        this.x = x1;  
        this.y = y1;  
    }  
    public Complex( Complex z ) {  
        this.x = z.x;  
        this.y = z.y;  
    }  
}
```

Constructor Methods

- Constructor เรียกกันเองได้ โดยใช้ this

```
public class Complex {  
    public double x, y;  
  
    public Complex( ) {  
        Complex(0, 0); // WRONG  
    }  
    public Complex( double x1, double y1 ) {  
        this( x1, y1 );  
    }  
    public class Complex {  
        public double x, y;  
    }  
    public Complex( ) {  
        this(0, 0); // CORRECT  
    }  
    public Complex( double x1, double y1 ) {  
        this.x = x1;  
        this.y = y1;  
    }  
}
```

เรื่องน่ารู้

- Utility class คือคลาสที่มีแต่ class methods (เช่น `Math.*`, `Arrays.*`, `JLabIO.*`)
- คลาสไม่จำเป็นต้องมีเมธ็อด main เมื่อเป็นคลาสที่ผู้ใช้ไม่ได้เรียกใช้งานโดยตรง
- ถ้าคลาสไม่มี constructor ระบบจะเติมให้แบบไม่มีการรับพารามิเตอร์
- ตั้งชื่อคลาสเป็น noun ตั้งชื่อเมธ็อดเป็น verb
- ตัด `this.` ทิ้งก็ได้

```
public class Complex {  
    public double x, y;  
    ...  
    public boolean equals( Complex z ) {  
        return x == z.x && y == z.y;  
    }  
}
```

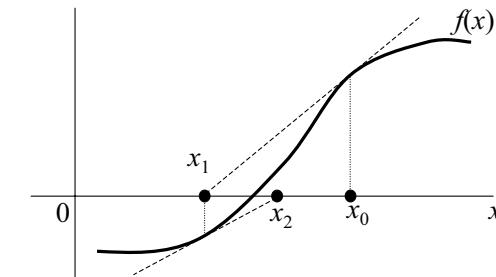
2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 29

ตัวอย่าง : หารากด้วย Newton-Raphson

- ต้องการหารากของสมการ $f(x) = 0$
- หาได้เฉพาะรากที่เป็นจำนวนจริง
- วิธีของ Newton-Raphson
 - ต้องรู้อนุพันธ์ของ $f(x)$
 - ถ้า x_k คือค่าประมาณของราก จะสามารถคำนวณค่าประมาณของราก (ที่ดีกว่า) x_{k+1} ดังนี้

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 30

Class NewtonRaphson

```
public class NewtonRaphson {  
    public static double f (double x) {  
        return x * x + 4 * x + 4;  
    }  
    public static double df (double x) {  
        return 2 * x + 4;  
    }  
    public static void solve (double xnew, double tol, int imax) {  
        double xold;  
        int i;  
        for (i = 0; i < imax; i++) {  
            xold = xnew;  
            xnew = xold - f(xold) / df(xold);  
            System.out.println("x = " + xnew);  
            if (Math.abs(xnew - xold) < tol) break;  
        }  
        if (i >= imax) System.out.println("not converge");  
    }  
    public static void main(String[] args) {  
        solve(0, 1e-10, 100);  
    }  
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 31

หาราก (เป็นจำนวนเชิงซ้อนได้)

- เขียนคลาส Complex ให้มี
 - data members : x และ y
 - constructor
 - object methods :
add, subtract, multiply, divide, abs, toString
- เขียนคลาส NewtonRaphsonComplex

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 32

Class Complex

```
public class Complex {  
    public double x, y;  
  
    public Complex(double x1, double y1) {  
        x = x1; y = y1;  
    }  
    public String toString() {  
        if (y >= 0) return "(" + x + " + " + i" + y + ")";  
        else return "(" + x + " - " + i" + (-y) + ")";  
    }  
    public Complex multiply(Complex z) {  
        return new Complex(x * z.x - y * z.y, x * z.y + y * z.x);  
    }  
    public Complex divide(Complex z) {  
        double abs2 = z.x * z.x + z.y * z.y;  
        return new Complex((x * z.x + y * z.y) / abs2,  
                           (z.x * y - z.y * x) / abs2);  
    }  
    public double abs() {  
        return Math.sqrt(x * x + y * y);  
    }  
    ...  
}
```

เขียนเมธ็อด add และ subtract เอง

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 33

Class NewtonRaphsonComplex

```
public class NewtonRaphsonComplex {  
  
    public static Complex f(Complex x) {  
        Complex two = new Complex(2, 0);  
        Complex c1 = x.multiply(x);  
        Complex c2 = x.multiply(two);  
        return c1.add(c2.add(two));  
    }  
  
    public static Complex df(Complex x) {  
        Complex two = new Complex(2, 0);  
        Complex c2 = x.multiply(two);  
        return c2.add(two);  
    }  
}
```

$$f(x) = x^2 + 2x + 2$$

$$f'(x) = 2x + 2$$

ยังมีต่อ...

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 34

Class NewtonRaphsonComplex

...เขียนต่อ

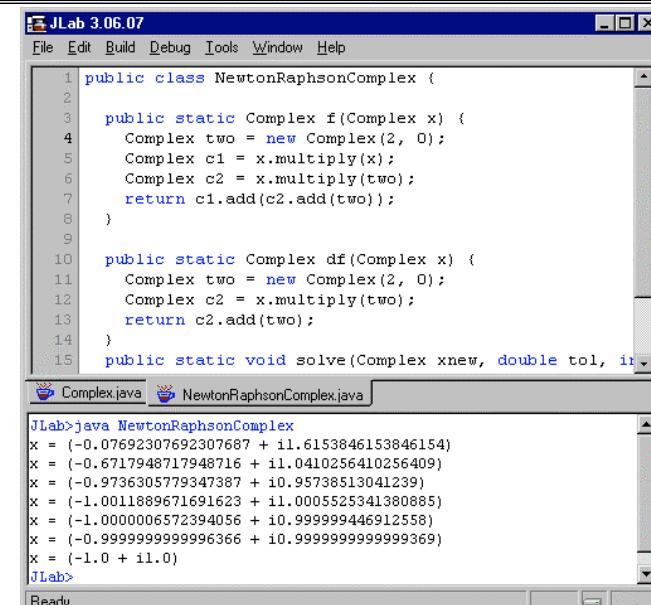
```
public static void solve(Complex xnew, double tol, int imax) {  
    Complex xold, temp;  
    int i;  
    for (i = 0; i < imax; i++) {  
        xold = xnew;  
        temp = f(xold).divide(df(xold));  
        xnew = xold.subtract(temp);  
        System.out.println("x = " + xnew);  
        temp = xnew.subtract(xold);  
        if (temp.abs() < tol) break;  
    }  
    if (i >= imax) System.out.println("not converge");  
}  
  
public static void main(String[] args) {  
    solve(new Complex(1, 3), 1e-10, 100);  
}
```

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 35

Edit + Compile + Run



JLab 3.0.07

```
File Edit Build Debug Tools Window Help  
1 public class NewtonRaphsonComplex {  
2     ...  
3     public static Complex f(Complex x) {  
4         Complex two = new Complex(2, 0);  
5         Complex c1 = x.multiply(x);  
6         Complex c2 = x.multiply(two);  
7         return c1.add(c2.add(two));  
8     }  
9     ...  
10    public static Complex df(Complex x) {  
11        Complex two = new Complex(2, 0);  
12        Complex c2 = x.multiply(two);  
13        return c2.add(two);  
14    }  
15    public static void solve(Complex xnew, double tol, int imax) {  
16        ...  
17        Complex xold, temp;  
18        int i;  
19        for (i = 0; i < imax; i++) {  
20            xold = xnew;  
21            temp = f(xold).divide(df(xold));  
22            xnew = xold.subtract(temp);  
23            System.out.println("x = " + xnew);  
24            temp = xnew.subtract(xold);  
25            if (temp.abs() < tol) break;  
26        }  
27        if (i >= imax) System.out.println("not converge");  
28    }  
29    public static void main(String[] args) {  
30        solve(new Complex(1, 3), 1e-10, 100);  
31    }  
32}
```

Complex.java NewtonRaphsonComplex.java

```
JLab>java NewtonRaphsonComplex  
x = (-0.07692307692307687 + i1.6153846153846154)  
x = (-0.6717948717948716 + i1.0410256410256409)  
x = (-0.9736305779347387 + i0.95738513041239)  
x = (-1.0011889671691623 + i1.0005525341380885)  
x = (-1.0000006572394056 + i0.999999446912558)  
x = (-0.999999999996366 + i0.999999999999369)  
x = (-1.0 + i1.0)  
JLab>  
Ready
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 36

คำถ้า

```
Complex c1 = new Complex(5, 0);
Complex c2 = new Complex(4, 0);
Complex c3 = new Complex(3, 0);

Complex c4 = c1.add(c2).subtract(c3);
Complex c5 = c1.multiply(c2).add(c3);
Complex c6 = c1.add(c2).multiply(c3);
Complex c7 = c1.add(c2.multiply(c3));
```

c4, c5, c6, c7 = ?

```
public Complex {
    ...
    public Complex sqr() {
        return [redacted];
    }
    ...
}
```

จะเขียนเมมท์ออดยกกำลังสองในคลาส
Complex โดยการใช้เมมท์ออด multiply

โปรแกรมแสดง Mandelbrot set

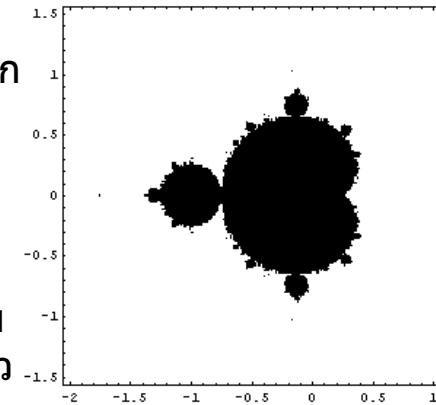
```
public void paint(Graphics g) {
    int n;
    Complex c, d;
    double stepX = 3.0 / getSize().width;
    double stepY = 3.0 / getSize().height;
    for (double y = -1.5; y <= 1.5; y += stepY) {
        for (double x = -2.0; x <= 1.0; x += stepX) {
            c = new Complex(x, y);
            d = new Complex(0, 0);
            n = 0;
            while (d.abs() < 2 && n++ < 256) {
                d = d.multiply(d).add(c);
            }
            if (n >= 256) {
                g.drawRect((int)((x + 2) / stepX),
                           (int)((y + 1.5) / stepY), 1, 1);
            }
        }
    }
}
```

|d_n| < 2

d_n = (d_{n-1})² + c

Mandelbrot Set (อีกครั้ง)

- ให้ลำดับของจำนวน เชิงซ้อน d_n คำนวณได้จาก $d_n = (d_{n-1})^2 + c$ โดยที่ $d_0 = 0+i0 = 0$ และ c เป็นจำนวนเชิงซ้อน



แกน x แทนเส้นจำนวนจริง แกน y แทนเส้นจำนวนจินตภาพ
จุดดำในรูปคือสมาชิกของ Mandelbrot set

แบบฝึกหัด : อย่างได้คลาส "Rational"

- Rational เป็นคลาสเก็บเลขตรรกยะ
 - ต.ย. 3/4, 15/7, 8/1
- มีสอง fields : numerator และ denominator
 - ต.ย. 15/7 มี numerator เป็น 15 และ denominator เป็น 7
- มีเมมท์ออดดังนี้
 - add, subtract, multiply, divide, toString, และ equals
- ควรเก็บแบบบลดรูป เช่น 4/8 ควรเก็บเป็น 1/2

```
public class Rational {
    public int numerator;
    public int denominator;
    ...
}
```

2110101 : การทำโปรแกรมคอมพิวเตอร์

Composition & Inheritance

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

เมื่อต้องการสร้างคลาสใหม่

- คิดถึง
 - คลาสเก่าที่เคยเขียน
 - คลาสมมาตรฐานของระบบจาวา
 - คันจาก j2sdk help file (มี ~ 3000 คลาส)
 - คลาสที่คนอื่นเขียนและอนุญาตให้ใช้
 - คันในอินเตอร์เน็ต (JNL, JSci, JLAPACK, JEdit, Lucene, ...)
 - คลาสที่ขายในห้องตลาด
- นำมาใช้ด้วยวิธี
 - copy & change source code (โนราน)
 - composition
 - inheritance

avoid reinventing the wheel

หัวข้อ

- Composition
- Inheritance
 - subclass และ superclass
 - Overriding vs. Overloading
 - class Object
 - Inheritance Hierarchy
 - upcasting และ downcasting
- Composition vs. Inheritance

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 2

เลือกของดี

- well defined
- carefully tested
- well documented
- portable
- highly rated
- open source

Reusability

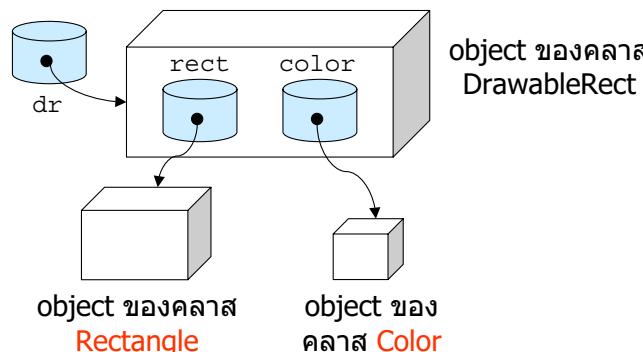
Rapid
Application
Development

Composition

- สร้างคลาสใหม่ที่ประกอบด้วย fields ที่เป็น objects ของคลาสอื่น
- เขียน methods ของคลาสใหม่ โดยใช้ fields และ methods ขององค์ประกอบภายในให้เป็นประโยชน์
- รายละเอียดของ methods ใหม่เป็นเสมือน "การ" ที่นำของที่มีอยู่มาปะติดกันให้ได้สิ่งประดิษฐ์ใหม่ๆ

Composition

- สร้างคลาสใหม่ประกอบด้วย fields
 - ออบเจกต์ของคลาส Rectangle
 - ออบเจกต์ของคลาส Color



อยากได้ "สีเหลี่ยมที่วาดได้"

- อยากได้คลาส DrawbleRect (มีสีด้วย) ซึ่งมีเมธอด
 - equals(DrawbleRect r) : ทดสอบว่า r เหมือน this ?
 - draw(Graphics g) : วาด this ?
- ~~เขียนใหม่หมด~~ (หารือ)
- ใช้คลาสมาตรฐานของjava Rectangle, Color
- Rectangle มีสมาชิก:
 - fields :
 - x, y, width, height
 - methods :
 - add, contains, equals, getWidth, getHeight, grow, intersection, intersects, move, toString, union, ...

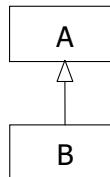
Composition

```
import java.awt.*;
public class DrawbleRect {
    public Rectangle rect;
    public Color color;
    public DrawbleRect(int x, int y, int w, int h, Color c) {
        this.rect = new Rectangle(x, y, w, h);
        this.color = c;
    }
    public boolean equals(DrawbleRect r) {
        if (r == null) return false;
        return this.rect.equals(r.rect) &&
            this.color.equals(r.color);
    }
    public void draw(Graphics g) {
        g.setColor(this.color);
        g.fillRect(this.rect.x, this.rect.y,
            this.rect.width, this.rect.height);
    }
}
```

ประกอบด้วยออบเจกต์ของคลาส Rectangle และ Color

Inheritance

- เขียนคลาสใหม่ (B) ชื่ง "ขยาย" คุณสมบัติจากคลาสเก่า (A) ที่มีอยู่ (**class B extends A**)
- เรียกคลาส B ว่า "รับมรดก" จากคลาส A
 - B มีทุกๆ fields ของ A (โดยอัตโนมัติ)
 - B มีทุกๆ methods ของ A (โดยอัตโนมัติ)
(ยกเว้น constructors)
 - เรียก B ว่าเป็น subclass ของ A
 - เรียก A ว่าเป็น superclass ของ B
 - ตีความໄດ้ว่า B เป็นคลาสที่เป็นกรณีพิเศษของ A
 - ขอบเขตของ B เป็น A
 - แต่ขอบเขตของ A ไม่จำเป็นต้องเป็น B
 - ตัวอย่างเช่น **class Cat extends Mammal**



Inheritance : แบบง่ายสุดๆ

- เขียนคลาสใหม่ เพียงแค่บอกว่าขยายจากคลาส

```
import java.awt.*;  
public class R1 extends Rectangle {  
}
```

- เหมือน **Rectangle** ทุกประการ (ยกเว้น constructors)
- R1 มี fields และ methods ต่างๆ ตามที่ **Rectangle** มี
- ไม่มี constructor อะไรเลย ระบบจะเดิน constructor แบบไม่รับพารามิเตอร์ให้

```
import java.awt.*;  
public class R1 extends Rectangle {  
    public R1() {  
        super();  
    }  
}
```

← เรียก constructor ของ superclass

Inheritance : แบบที่ควรทำ

- กำหนดให้คลาสใหม่ขยายจากคลาสเก่าที่มี
- เพิ่ม fields ที่จำเป็น
- เขียน constructors ที่อยากให้บริการ
- เพิ่ม methods ใหม่ๆ ที่ superclass ไม่มี
- เปลี่ยน methods เดิมของ superclass ให้มีพฤติกรรมใหม่ (เรียกว่า overriding method)
- จาวาไม่อนุญาตให้ลบ methods ของ superclass ที่ subclass ไม่อยากได้

Inheritance

```
import java.awt.*;  
  
public class DrawableRect extends Rectangle {  
    public Color color;  
  
    public DrawableRect(int x, int y, int w, int h, Color c) {  
        super(x, y, w, h);  
        this.color = c;  
    }  
  
    public boolean equals(Object r) {  
        if ( !super.equals(r) ) return false;  
        if ( this.getClass() != r.getClass() ) return false;  
        return this.color.equals(((DrawableRect) r).color);  
    }  
  
    public void draw(Graphics g) {  
        g.setColor(this.color);  
        g.fillRect(super.x, super.y, super.width, super.height);  
    }  
}
```

← เพิ่ม field ใหม่

← เขียน constructor

← เปลี่ยน method เดิม

← เพิ่ม method ใหม่

การใช้ fields ของ superclass

- อ้างอิง fields ของ superclass ได้โดยใช้ **super**. นำหน้าชื่อ field
- หรือจะใช้ **this**. นำหน้าชื่อ field ของ superclass ก็ได้ (เพราะ "รับมรดก" มาแล้ว)
- หรือจะใช้แค่ชื่อ field ก็ไม่ผิด (เพราะตัด **this**. ได้)

```
public void draw(Graphics g) {  
    g.setColor(this.color);  
    g.fillRect(super.x, super.y, super.width, super.height);  
}  
  
public void draw(Graphics g) {  
    g.setColor(this.color);  
    g.fillRect(this.x, this.y, this.width, this.height);  
}  
  
public void draw(Graphics g) {  
    g.setColor(this.color);  
    g.fillRect(x, y, width, height);  
}
```

เขียนแบบนี้
รู้ที่มา

การใช้ methods ของ superclass

- อ้างอิง methods เดิม ของ superclass ได้โดยใช้ **super**. นำหน้าชื่อ method
- หรือจะใช้ **this**. นำหน้าชื่อ method (ที่ไม่ได้ถูก overridden) ของ superclass ก็ได้ (เพราะ "รับมรดก" มาแล้ว)

```
public boolean equals(Object r) {  
    if ( !super.equals(r) ) return false;  
    if ( this.getClass() != r.getClass() ) return false;  
    return this.color.equals(((DrawableRect) r).color);  
}
```

เรียก equals ของ superclass เพื่อทดสอบว่าเป็น **Rectangle** ที่
เหมือนกันหรือไม่ (ตรงนี้จะใช้ **this.equals** ไม่ได้ เพราะ.....)

แบบฝึกหัด

- เขียนเมธ็อด **toString** ให้กับ **DrawableRect**
- ข้อแนะนำ : ควรใช้ **toString** ของ superclass

```
public class DrawableRect extends Rectangle {  
    ...  
    public String toString() {  
        ...  
    }  
}
```

Overriding และ Overloading

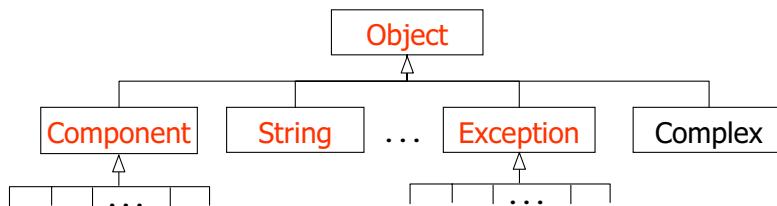
- Overriding เป็นการเขียนเมธ็อดใน subclass ซึ่งมีชื่อ และรายการพารามิเตอร์ ประก徂อยู่แล้ว ใน superclass (เป็นการ "ลับล้าง" ของเก่าที่มีอยู่)
- Overloading เป็นการเขียนเมธ็อดใหม่ ซึ่งมีชื่อซ้ำ แต่รายการพารามิเตอร์ไม่เหมือนกับของเดิม (เป็นการ "เพิ่มภาระ" ให้กับชื่อเมธ็อดว่าทำงานหลาย แบบ ขึ้นกับรายการพารามิเตอร์ที่ได้รับ)

```
public class A {  
    ...  
    public int f() {  
        ...  
    }  
}
```

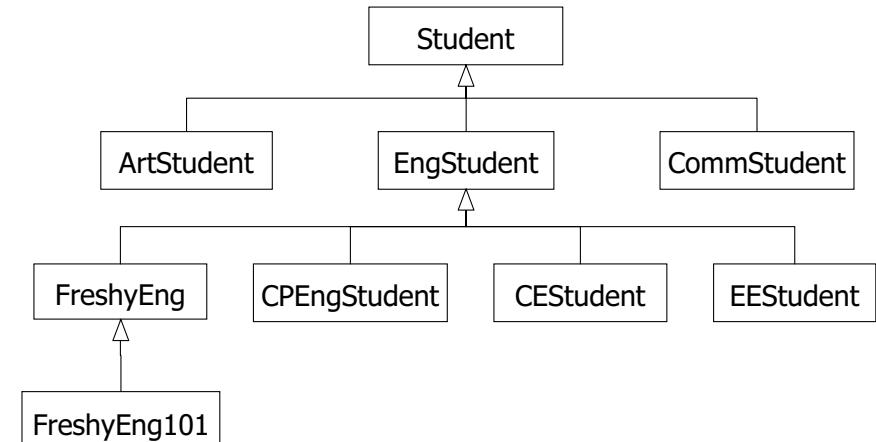
```
public class B extends A {  
    public int f() {           // overriding  
        ...  
    }  
    public int f(int x) {     // overloading  
        ...  
    }  
}
```

คลาส Object

- จะมีคลาสมาตรฐานชื่อว่า **Object** (ขยาย ชื่อ **Object** นี้เป็นชื่อคลาส ขึ้นต้นด้วยโอลิฟ)
- คลาส **Object** เป็น "บรรพบุรุษ" ของทุกคลาสในjava
- การเขียนคลาสใหม่ที่ไม่ได้ **extends** จากคลาสใด ถือได้ว่า **extends** จากคลาส **Object**



Inheritance Hierarchy



การแสดง Inheritance ใน Help File

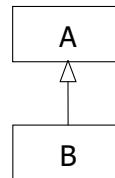
```
java.lang.Object
|
+-JSci.maths.Matrix
|
+-JSci.maths.ComplexMatrix
|
+-JSci.maths.ComplexSquareMatrix
|
+-JSci.maths.ComplexTridiagonalMatrix
|
+-JSci.maths.ComplexDiagonalMatrix
```

แบบฝึกหัด

- เขียน inheritance hierarchy ของคลาสต่อไปนี้
 - Quadrilateral, Trapesoid, Parallelogram, Rectangle, และ square
 - Vehicle, Car, Truck, Sedan, Coupe, PickupTruck, SportUtilityVehicle, Minivan, Bicycle, และ Motocycle
 - Archosaurs, Thecodonts, Pterosaurs, Dinosaurs, Crocilians, Saurischians, และ Ornithischians

Casting

- กำหนดให้ B **extends A**
 - B รับพฤติกรรมทุกอย่างจาก A
 - B "เป็น" A (คือเป็นกรณีพิเศษของ A)
- **Upcasting**
 - เราสามารถ cast ครอบเจกต์ของ B ให้เป็น A ได้
 - upcast ได้เสมอ
- **Downcasting**
 - ครอบเจกต์ของ B ที่เดยถูก upcast ให้เป็น A สามารถถูก cast กลับลงมาเป็น B ได้
 - ถ้า downcast ไม่ได้จะเกิด **ClassCastException**



แบบฝึกหัด

- เขียนคลาส Square โดยใช้คลาส **Rectangle** แบบ
 - Composition
 - Inheritance
- **Fields :**
 - x, y ระบุพิกัดตรงกลางสีเหลืองจัตุรัส
 - length ระบุความยาวด้าน
- **Methods :**
 - `boolean contains(int x, int y)`
 - `void translate(int dx, int dy)`
 - `int getLength()`
 - `int setLength()`
 - `boolean equals(Object s)`
 - `String toString()`

อ่านความหมายของ
แต่ละเมธ็อดได้จาก
JDK help file ของ
Rectangle

มาตรฐาน equals ของ DrawableRect

- equals รับออบเจกต์ของคลาส **Object** ดังนี้
 - equals นี้ override equals ของ **Rectangle**
- ใช้ `super.equals(r)`
 - เพื่อทดสอบว่าเป็น **Rectangle** หรือไม่
- ใช้ `this.getClass()` กับ `r.getClass()`
 - เพื่อเบริยบเทียบว่า r เป็นคลาสประเภทเดียวกับ `this` หรือไม่
 - ถ้าเป็นคลาสประเภทเดียวกันจะได้ downcast จาก **Object** มาได้ เพื่อจะได้หยิบ color : `((DrawableRect) r).color`

```
public boolean equals(Object r) {  
    if ( !super.equals(r) ) return false;  
    if ( this.getClass() != r.getClass() ) return false;  
    return this.color.equals(((DrawableRect) r).color);  
}
```

Composition vs. Inheritance

- ใช้ composition เมื่อเราใช้คุณสมบัติของคลาสอื่น มาเป็นส่วนประกอบเพื่อออกแบบของใหม่
 - เช่น รถ "มี" ล้อ เครื่องยนต์ หน้าต่าง ประตู
 - ดังนั้นคลาส Car ประกอบด้วย fields ของครอบเจกต์จาก คลาส Wheel, Engine, Window, Door เป็นต้น
- ใช้ inheritance เมื่อเราต้องการคลาสใหม่ซึ่งเป็น กรณีพิเศษของคลาสเดิม จึงต้องการคุณสมบัติเดิม ส่วนใหญ่ของคลาสเก่า มีการเปลี่ยนหรือเพิ่มเติม คุณสมบัติใหม่บ้าง
 - เช่น นิสิตวิศวฯ จุฬาฯ "เป็น" นิสิตจุฬาฯ แต่มีอะไรบางอย่างเพิ่มเติมจากกรณีทั่วไป
 - ดังนั้นคลาส EngCUSTudent **extends** CUStudent

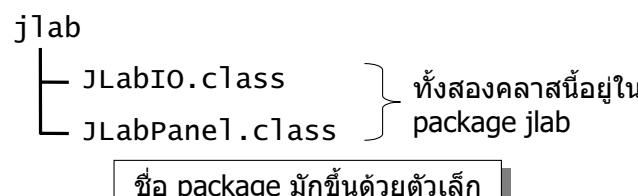
2110101 : การทำโปรแกรมคอมพิวเตอร์

เรื่องต้องรู้ (บัง)

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

Package

- โปรแกรมJAVAประกอบด้วยคลาสมากมาย ทั้งที่เขียนเอง ของระบบJAVA หรือของคนอื่น
- จะจัดเก็บคลาสต่างๆ อย่างไร ที่ไหน และมีวิธีป้องกันชื้อช้าได้อย่างไร
- จัดเก็บคลาสต่างๆ ในรูปของ package
- คลาสที่อยู่ใน package เดียวกัน คืออยู่ใน directory เดียวกัน

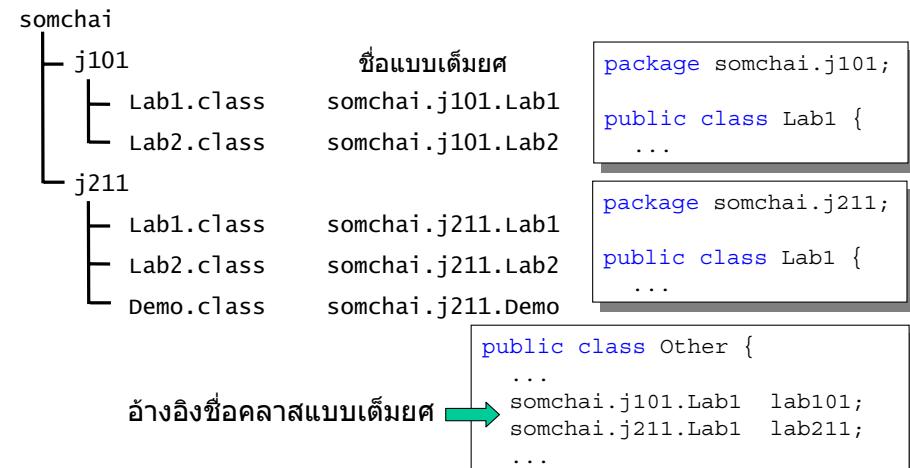


หัวข้อ

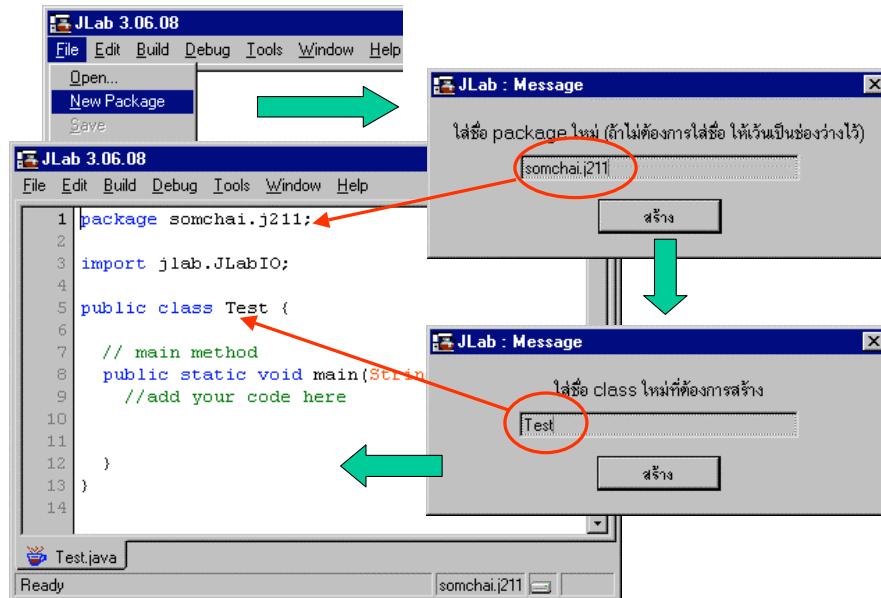
- packages, import
- private, public, final, static
- Error & Exception Handling (Optional)

Package

- Package มีลำดับชั้น : มี subdirectory ได้
- ต้องระบุชื่อ package ในโปรแกรมด้วย



การสร้าง Package ใน JLab



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 5

import

- การใช้คลาสที่อยู่นอก package ของตัวเอง
 - เขียนชื่อ class รวมทั้ง package ให้ครบถ้วน
 - หรือใช้ **import** ก่อน แล้วอ้างเฉพาะชื่อคลาสได้เลย
 - import แบบระบุชื่อคลาส
 - import แบบเหมาหมดทุกคลาส

```
import jlab.JLabIO;
import jlab.*;

public class Happy {
    public void A() {
        int x = JLabIO.readInt();
    }
}

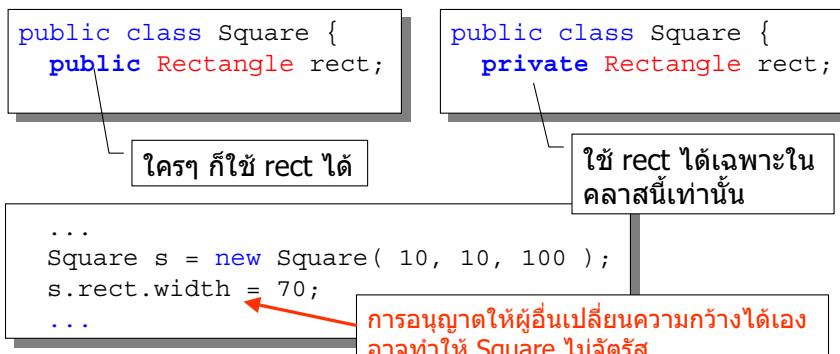
public class Happy {
    public void A() {
        int x = jlab.JLabIO.readInt();
    }
}
```

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 6

การควบคุมการใช้ fields และ methods

- ใช้ **public** กำกับ
 - เปิดเผย : คลาสนี้ๆ ใช้ได้หมด
- ใช้ **private** กำกับ
 - ปกปิด : ใช้ได้เฉพาะภายในคลาสเจ้าของเท่านั้น



2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 7

ข้อแนะนำ

- ให้ทุกๆ fields เป็น **private**
- มี **public method** ให้หยิบหรือเปลี่ยนค่าของ field ได้

```
public class Square {  
    private Rectangle rect;  

  
    public Square(int x, int y, int len) {  
        this.rect = new Rectangle(x, y, len, len);  
    }  
    public int getLength() {  
        return this.rect.width;  
    }  
    public void setLength(int len) {  
        if (len > 0)  
            this.rect.width = this.rect.height = len;  
    }  
}
```

ปลอดภัย และตรวจสอบการทำงานของโปรแกรมได้ง่ายขึ้น

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 8

ยังมีแบบอื่น

- นอกจาก **private** และ **public** แล้วยังมี
 - protected** อนุญาตให้เฉพาะคลาสใน package เดียวกัน และ subclass เท่านั้นที่ใช้ **protected** ของคลาสอื่น
 - ไม่ต้องใส่อะไรกับ อนุญาตให้เฉพาะคลาสใน package เดียวกันเท่านั้น

```
package mars;
public class A {
    protected int count;
    ...
}
package earth;
public class B extends mars.A {

    public void retire() {
        super.count = 5; // OK
        C c = new C();
        c.number = 7; // fail
        ...
    }
}
```

final

- เติม **final** หน้า field
 - ไม่ต้องการให้เปลี่ยนแปลงค่าได้ (constant)
- เติม **final** หน้า method
 - ไม่ต้องการให้ subclass override ได้
- เติม **final** หน้า class
 - ไม่ต้องการให้ใคร extends ได้

มักตั้งชื่อ final field ด้วยตัวอักษรตัวใหญ่ทุกตัว

```
final String COMPUTER = "cp";
final String ELECTRICAL = "ee";
final String CIVIL = "ce";
...
```

static

- เติม **static** หน้า method ทำให้เป็น class method
- เติม **static** หน้า field ทำให้เป็น class data
 - ที่เก็บไม่ได้อยู่ประจำแต่ละออบเจกต์ที่สร้าง
 - แต่อยู่ประจำคลาสเจ้าของ ดังนั้นจึงมีเพียงที่เก็บเดียว

```
public class Undergrad {
    public static final int FRESHY = 1;
    public static final int SOPHOMORE = 2;
    public static final int JUNIOR = 3;
    public static final int SENIOR = 4;
    private int status;
    public void setStatus( int s ) {
        switch( s ) {
            ...
            this.status = s;
            ...
        }
    }
}
```

```
Undergrad s = new Undergrad();
s.setStatus( Undergrad.FRESHY );
...
```

Error และ Exception

- การทำงานของโปรแกรมมีโอกาสเกิดความผิดพลาด หรือพบสิ่งผิดปกติ
- ความผิดปกติอาจเกิดที่
 - ภายในตัวระบบจาวา รวมทั้งภายในเมทธอดของคลาสมادرฐานของจาวา
 - หน่วยความจำหมด, อ้างอิง index ของอาร์เรย์นอกช่วงที่จะองไว้, ...
 - ภายในเมทธอดของผู้เขียนเอง
 - ข้อมูลในตัวแปร ไม่เป็นไปตามที่หวัง (เช่นต้องไม่ติดลบ, ...)

Exceptions ที่เกิดขึ้นจากระบบจาวา

• การทำงานมีโอกาสผิดปกติ (Exception)

```
// ArithmeticException  
int a = 4, b = 0;  
int c = a / b;
```

```
// ArrayIndexOutOfBoundsException  
int [] a = int[40];  
a[40] = 20;
```

```
// StackOverflowError  
void Jeng() {  
    Jeng();  
}
```

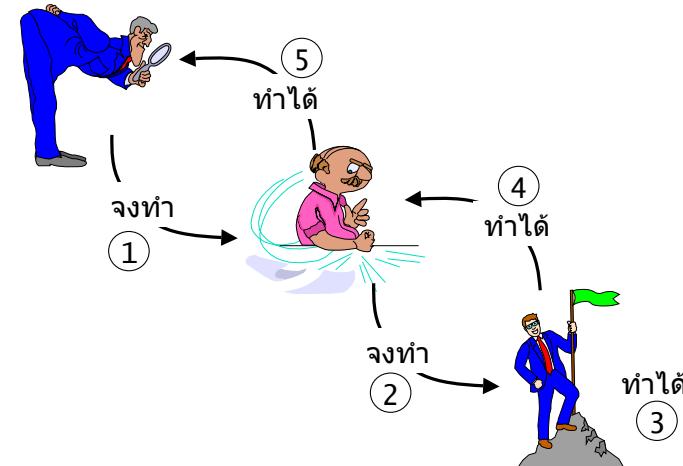
```
// ClassCastException  
Object a = new Rectangle();  
System.out.println((String) a);
```

```
// NegativeArraySizeException  
int n = -100;  
Vector [] a = new Vector[n];
```

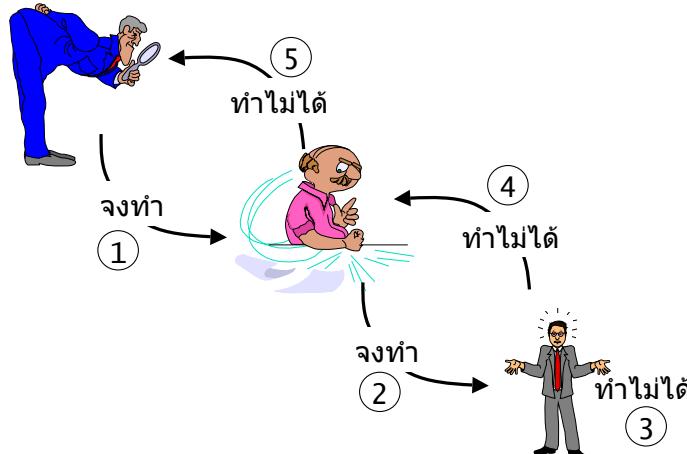
```
// ArrayStoreException  
Object [] a = new String[5];  
a[0] = new Rectangle();
```

```
// FileNotFoundException  
BufferedReader in = JLabIO.openFile( "C:\\\\DATA.IN" );  
String txt = in.readLine();
```

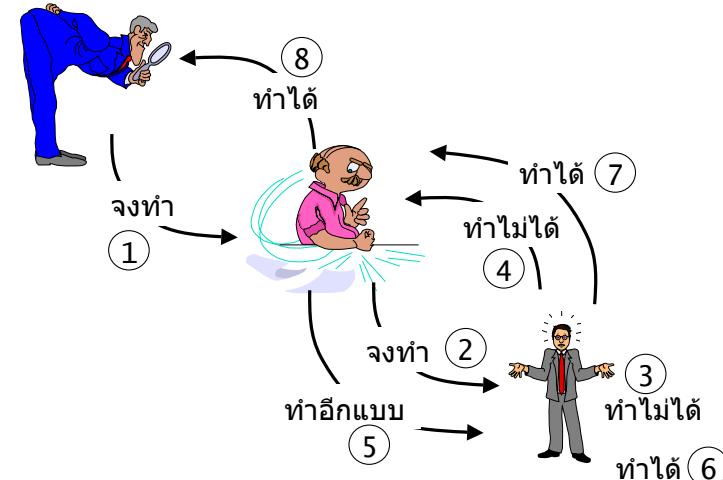
การเรียกเมธอด



ผิดปกติแล้วไม่จัดการ



ผิดปกติต้องจัดการ



Robustness

- ถ้าเกิดความผิดปกติ แล้วไม่มีส่วนใดของโปรแกรมที่รองรับสถานะการณ์ที่เกิดขึ้น
 - ระบบจะแสดงข้อความอธิบายความผิดปกติออกจอภาพ
 - แล้วเลิกการทำงาน
- ระบบที่แข็งแกร่งควรจัดการกับความผิดปกติที่อาจเกิดขึ้นให้มากที่สุด โดยทำให้โปรแกรมยังทำงานต่อไปได้
- ตัวอย่าง : ต้องการเปิดแฟ้มข้อมูล แต่หาไม่พบ
 - แทนที่จะบอกผู้ใช้ว่า "file not found" แล้วเลิกการทำงาน
 - ก็บอกผู้ใช้ให้ใส่ชื่อแฟ้มใหม่ และลองเปิดแฟ้มใหม่

Throwing Exceptions

- เมื่อเกิดความผิดปกติภายในเมธ็อด และต้องการแจ้งให้เมธ็อดที่เรียกมาให้ทราบ
 - สร้างออบเจกต์ของคลาสที่แทนความผิดปกติที่เกิดขึ้น
 - แล้ว "โยน" ออบเจกต์นั้นให้ผู้เรียกรับทราบ
- ที่หัวเมธ็อดต้องระบุด้วยว่า เมธ็อดนี้อาจมีการ "โยน" สิ่งผิดปกติประเภทใดบ้าง

```
public double [] qRoot( double a, double b, double c )  
throws IllegalArgumentException {  
    double t = b * b - 4 * a * c;  
    if ( t < 0 ) {  
        IllegalArgumentException e =  
            new IllegalArgumentException("imaginary root");  
        throw e;  
    }  
    ...  
}
```

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Exception Classes

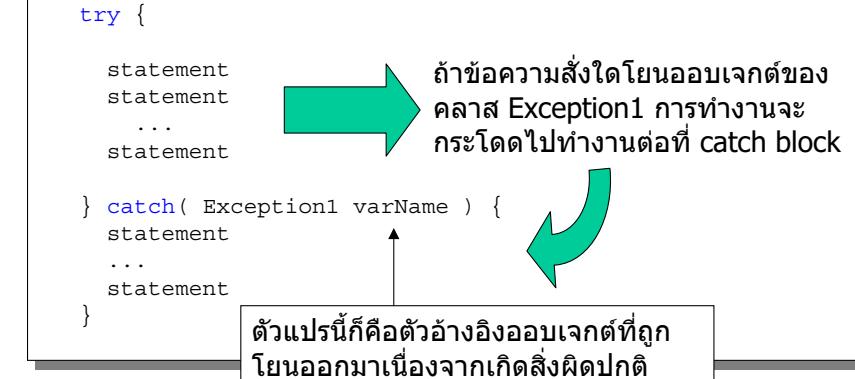
- java ใช้คลาสเพื่อเขียนบรรยายความผิดปกติ
- ชื่อคลาสจะสื่อความหมายถึงประเภทของความผิด
 - IOException, OutOfMemoryError, ...

```
package java.lang;  
  
public class IllegalArgumentException extends RuntimeException {  
    public IllegalArgumentException() {  
        super();  
    }  
    public IllegalArgumentException(String s) {  
        super(s); // s is detail exception message  
    }  
}
```

คลาส **IllegalArgumentException** มีแค่นี้เอง คลาสนี้มักใช้แทนความผิดปกติกรณีที่เมธ็อดได้รับข้อมูลมาไม่ตรงตามที่ได้ตกลงกันไว้

Catching Exception

- เมื่อใช้อะไรที่อาจ "โยน" สิ่งผิดปกติ
- ก็ต้องค่อย "รับ" สิ่งผิดปกติตัวย
- ใช้คำสั่ง try-catch



มีได้หลาย catch blocks

```
try {  
    statement  
    ...  
    statement  
}  
catch( Exception1 varName ) {  
    statement  
    ...  
    statement  
}  
catch( Exception2 varName ) {  
    statement  
    ...  
    statement  
}  
catch( Exception3 varName ) {  
    statement  
    ...  
    statement  
}
```

การทำงานจะกระโดดไปทำงานต่อที่ catch block ใด ก็ขึ้นกับว่าอุปนิสัยที่ถูกโยนมาตรงกับคลาสของ catch block ใด

ตัวอย่าง

- เมธอด dumpFile อาจโยน IOException object

```
public void dumpFile(String fn) throws IOException {  
    BufferedReader in = JLabIO.openFile(fn);  
    String txt;  
    while ( (txt = in.readLine()) != null ) {  
        System.out.println(txt);  
    }  
    in.close();  
}
```

```
do {  
    try {  
        String fn = JLabIO.readString("file = ");  
        dumpFile(fn);  
        success = true;  
    } catch( IOException e ) {  
        System.err.println(e.getMessage() + " : try again");  
        success = false;  
    }  
} while ( !success );
```

ยังมีเรื่องจุกจิกอีกมาก

- ถ้าไม่ "รับ" ก็ต้อง "โยน" ต่อ
- คลาสของความผิดปกติทั้งหลาย extends มาจากคลาスマatrฐาน 3 ประเภทคือ
 - Error, Exception และ RuntimeException
- ลำดับของ catch block มีความหมาย
- ไม่ต้อง catch ความผิดปกติที่ extends จาก RuntimeException ก็ได้
- ...

2110101 : การทำโปรแกรมคอมพิวเตอร์

การโปรแกรมแบบเวียนเกิด

ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อ

- Recurrences
- Iterative vs. Recursive
- Examples

2110101 วิศวกรรมคอมพิวเตอร์ จุฬา

7/8/2002 2

ความสัมพันธ์เวียนเกิด (Recurrences)

- การเขียนความสัมพันธ์ของจำนวนเต็มในลำดับ

$$0, 1, 2, 3, 4, \dots \quad a_n = a_{n-1} + 1 \quad \text{เมื่อ } n > 0, \quad a_0 = 0$$

$$3, 5, 7, 9, \dots \quad a_n = a_{n-1} + 2 \quad \text{เมื่อ } n > 0, \quad a_0 = 3$$

$$0, 1, 3, 6, 10, 15, \dots \quad a_n = a_{n-1} + n \quad \text{เมื่อ } n > 0, \quad a_0 = 0$$

$$0, 1, 1, 2, 3, 5, 8, \dots \quad a_n = a_{n-1} + a_{n-2} \quad \text{เมื่อ } n > 1, \\ a_0 = 0, \quad a_1 = 1$$

0, 1, 3, 6, 10, 15, ... (Iterative)

- เขียนโปรแกรมหาตัวที่ n
- รู้ว่า $a_n = (0+1+2+\dots+n)$

```
import jlab.JLabIO;

public class Sum2N {
    public static void main(String[] args) {
        int n = JLabIO.readInt("n = ");
        System.out.println(a(n));
    }
    static int a(int n) {
        int s = 0;
        for (int i = 0; i <= n; i++)
            s += i;
        return s;
    }
}
```

0, 1, 3, 6, 10, 15, ... (Recursive)

- เขียนโปรแกรมหาตัวที่ n
- รู้ว่า $a_n = a_{n-1} + n$ เมื่อ $n > 0$, $a_0 = 0$

```
import jlab.JLabIO;
public class Sum2N {
    public static void main(String[] args) {
        int n = JLabIO.readInt("n = ");
        System.out.println(a(n));
    }
    static int a(int n) {
        if (n <= 0)
            return 0;
        else
            return a(n-1) + n;
    }
}
```

โปรแกรมเรียก
(recursive program)

$$a_n = \begin{cases} 0 & \text{if } n \leq 0 \\ a_{n-1} + n & \text{otherwise} \end{cases}$$

0, 1, 1, 2, 3, 5, 8, 13, ... (Iterative)

- เขียนโปรแกรมหาตัวที่ n

```
import jlab.JLabIO;
public class Fibo {
    public static void main(String[] args) {
        int n = JLabIO.readInt("n = ");
        System.out.println(a(n));
    }
    static int a(int n) {
        int p = 0, q = 1, t = n;
        for (int i = 2; i <= n; i++) {
            t = p + q;
            p = q;
            q = t;
        }
        return t;
    }
}
```

0, 1, 1, 2, 3, 5, 8, 13, ... (Recursive)

- เขียนโปรแกรมหาตัวที่ n
- รู้ว่า $a_n = a_{n-1} + a_{n-2}$ เมื่อ $n > 1$, $a_0 = 0$, $a_1 = 1$

```
import jlab.JLabIO;
public class Fibo {
    public static void main(String[] args) {
        int n = JLabIO.readInt("n = ");
        System.out.println(a(n));
    }
    static int a(int n) {
        if (n < 2) return n;
        return a(n-1) + a(n-2);
    }
}
```

$$a_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ a_{n-1} + a_{n-2} & \text{otherwise} \end{cases}$$

โปรแกรมหาผลรวมในอารเรย์ (Iterative)

- ต้องการ $d[0] + d[1] + d[2] + \dots + d[d.length-1]$

```
public static int sum(int [] d) {
    int s = 0;
    for (int i = 0; i < d.length; i++)
        s += d[i];
    return s;
}
```

โปรแกรมหาผลรวมในอารเรย์ (Recursive)

- ต้องการ $d[0] + d[1] + d[2] + \dots + d[d.length-1]$
- กำหนดให้ $s_n = d[0] + d[1] + \dots + d[n-1]$
- แสดงว่า $s_{n-1} = d[0] + d[1] + \dots + d[n-2]$
- จะได้ว่า $s_n = s_{n-1} + d[n-1]$ เมื่อ $n > 0$, $s_0 = 0$
- ขอเขียนเป็น
 $s(d,n) = s(d,n-1) + d[n-1]$ เมื่อ $n > 0$, $s(0) = 0$

```
public static int sum(int[] d) {  
    return sum(d, d.length);  
}  
private static int sum(int[] d, int n) {  
    if (n == 0) return 0;  
    return sum(d, n - 1) + d[n - 1];  
}
```

โปรแกรมหาค่ามากสุดในอารเรย์

- กำหนดให้ $m_n = \max(d[0], d[1], \dots, d[n-1])$
- แสดงว่า $m_{n-1} = \max(d[0], d[1], \dots, d[n-2])$
- ได้ $m_n = \max(m_{n-1}, d[n-1])$ เมื่อ $n > 0$, $s_1 = d[0]$
- ขอเขียน $m(d,n) = \max(m(d,n-1), d[n-1])$ $n > 0$, $s(1) = d[0]$

```
public static int max(int[] d) {  
    return max(d, d.length);  
}  
private static int max(int[] d, int n) {  
    if (n == 1) return d[0];  
    int m = max(d, n - 1);  
    return (m > d[n - 1] ? m : d[n - 1]);  
}
```

โปรแกรมหาตำแหน่งของค่ามากสุดในอารเรย์

```
public static int max(int[] d) {  
    return max(d, d.length);  
}  
private static int max(int[] d, int n) {  
    if (n == 1) return d[0];  
    int m = max(d, n - 1);  
    return (m > d[n - 1] ? m : d[n - 1]);  
}  
  
public static int maxI(int[] d) {  
    return maxI(d, d.length);  
}  
private static int maxI(int[] d, int n) {  
    if (n == 1) return 0;  
    int i = maxI(d, n - 1);  
    return (d[i] > d[n - 1] ? i : n - 1);  
}
```

คืนค่ามากสุด

คืน index i ที่
 $d[i]$ มีค่ามากสุด

การเรียงลำดับแบบเลือก

- มองอารเรย์เป็นสองส่วน
 - ส่วนขวา เรียงเสร็จแล้ว
 - ส่วนซ้าย ยังไม่เรียง
- ส่วนซ้าย
 - หาตัวที่มีค่ามากสุด
 - สลับกับตัวสุดท้าย
- หลังสลับ
 - ส่วนซ้ายลดหนึ่งตัว
 - ส่วนขวาเพิ่มหนึ่งตัว
 - แล้วก็ไปเรียงลำดับส่วนทางซ้ายต่อไป

23	2	3	14	5	99	9
23	2	3	14	5	9	99
23	2	3	14	5	9	99
9	2	3	14	5	23	99
9	2	3	14	5	23	99
9	2	3	5	14	23	99
9	2	3	5	14	23	99
5	2	3	9	14	23	99
5	2	3	9	14	23	99
5	2	3	9	14	23	99
3	2	5	9	14	23	99
3	2	5	9	14	23	99
2	3	5	9	14	23	99

การเรียงลำดับแบบเลือก

```
public static void selectionSort(int[] data) {  
    int i, t;  
    for (int k = data.length; k > 1; k--) {  
        i = maxI(data, k);  
        t = data[i];  
        data[i] = data[k - 1];  
        data[k - 1] = t;  
    }  
  
    public static void selectionSort(int[] data) {  
        selectionSort( data, data.length );  
    }  
    private static void selectionSort(int[] data, int n) {  
        int i, t;  
        if (n < 2) return;  
        i = maxI(data, n);  
        t = data[i];  
        data[i] = data[n-1];  
        data[n-1] = t;  
        selectionSort( data, n-1 );  
    }  
}
```

Iterative

Recursive

การคำนวณ $a^p \bmod m$

- $a^p \bmod m$ เป็นการคำนวณที่ใช้บ่อยในการเข้ารหัสลับ
- ตัวอย่างที่ 1 : $2^{20} \% 31 = ?$
 - คำนวณ 2^{20} ได้ 1048576 จากนั้น % 31 ได้ 1
- ตัวอย่างที่ 2 : $2^{100} \% 31 = ?$
 - คำนวณ 2^{101} ได้ 2535301200456458802993406410752 จากนั้น % 31 ได้ 2
 - ทำอีกแบบ : ใช้ $2^{20} \% 31$ ให้เป็นประโยชน์
 - $2^{101} \% 31 = 2 \times (2^{20})^5 \% 31$
 - ใช้ความรู้ที่ว่า $(ab \% m) = (a \% m)(b \% m)$
 - $2^{101} \% 31 = 2 \times (2^{20})^5 \% 31 = (2 \% 31)(2^{20} \% 31)^5 = 2$

การคำนวณ $a^p \bmod m$

$$a^p \% m = \begin{cases} 1 & \text{if } p = 0 \\ (a^{p/2} \% m)^2 \% m & \text{if } p \text{ is even} \\ 2(a^{(p-1)/2} \% m)^2 \% m & \text{if } p \text{ is odd} \end{cases}$$

```
public static int powerMod(int a, int p, int m) {  
    if (p == 0) return 1;  
    if (p == 1) return a % m;  
    int k = powerMod(a, p / 2, m);  
    k = (k * k) % m;  
    if ((p % 2) == 1) k = (2 * k) % m; // p is odd  
    return k;  
}
```

โปรแกรมแบบเวียนเกิด

- โปรแกรมที่มีการเรียกตัวเอง
- การทำงานแบ่งเป็นกรณีๆ
 - กรณีพื้นฐาน ทำเสร็จได้ทันที
 - กรณีอื่น เรียกตัวเอง โดยขนาดของปัญหาต้องเล็กลง
- เขียนง่าย (?) สวย (?) แต่ทำงานช้ากว่า