

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**
 - คำสั่ง JLab I/O

ข้อมูลประเภทอักขระ (**char**)

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

การแทนอักขระด้วยรหัส

- จาวาเก็บอักขระต่างๆ 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), '๐' (รหัส 0E58_H), 'ก' (รหัส 0E01_H), 'ข' (รหัส 0E02_H)
- รองรับสารพัดภาษา (internationalization เขียนย่อว่า i18n)

การเขียนค่าคงตัวแบบ `char`

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

2 ไม่เหมือนกับ '2'

- จำนวนเต็ม 2 `short`

00	02
----	----

 - ใช้กับการประมวลผลเชิงคณิตศาสตร์ (+ - * / % ...)
- อักขระ '2' `char`

00	32
----	----

 - ใช้สื่อสารระหว่างอุปกรณ์ แสดงทางจอภาพ หรือประมวลผลเชิงอักขระอื่นๆ
 - การประมวลผลเชิงคณิตศาสตร์กับอักขระ จะไม่ได้ผลอย่างที่คิด ('2' + '3' ไม่ได้ '5')
 - ถ้าจะนำไปประมวลผลเชิงคณิตศาสตร์ก็ต้องเปลี่ยนเป็นจำนวนเต็มเสียก่อน

ข้อสังเกต : มีจำนวนเต็ม 2 แต่ไม่มี character '-2'

ข้อมูลประเภทสายอักขระ (**String**)

- **String** เขียนเริ่มด้วย **S** ตัวใหญ่
- ข้อมูลแบบ **String** เก็บตัวอักขระได้หลายๆ ตัว
- ค่าคงตัวแบบ **String** เขียนภายในเครื่องหมาย "
- เขียนค่าคงตัวแบบ **String** ห้ามบรรทัดไม่ได้
- เรียกจำนวนอักขระว่า 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";
    }
}
```

ตัวปฏิบัติการ + กับ **String**

- **String** มีตัวปฏิบัติการตัวเดียวคือ +
- + คือการต่อสายอักขระสองสายเข้าด้วยกัน
- เมื่อข้อมูลแบบ **String** + กับข้อมูลใด ข้อมูลนั้น จะถูกเปลี่ยนเป็น **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
```

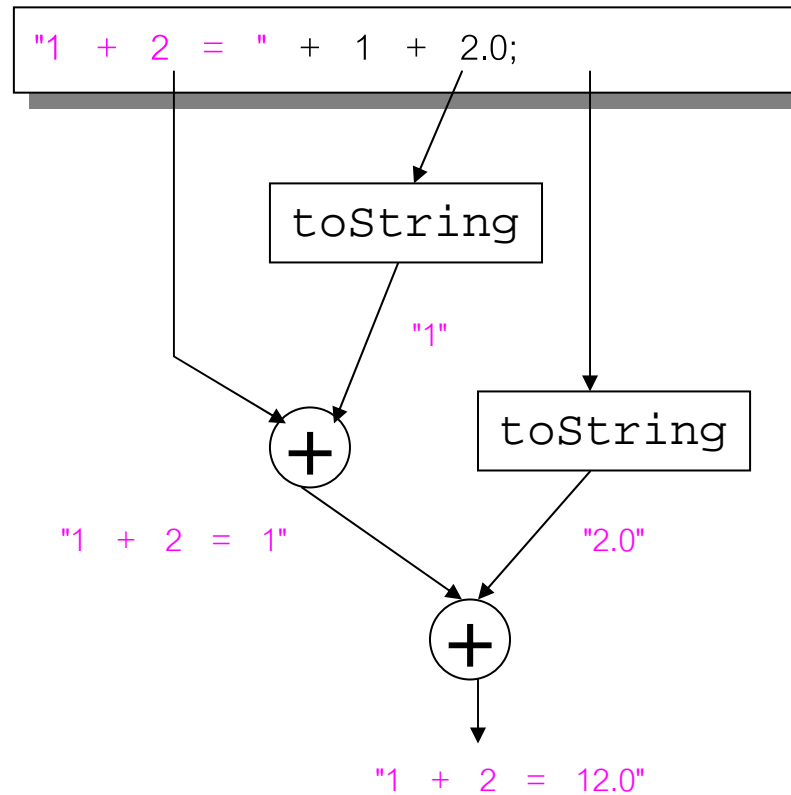
```
Approx. value of PIis
```

```
Approx. value of PI is
```

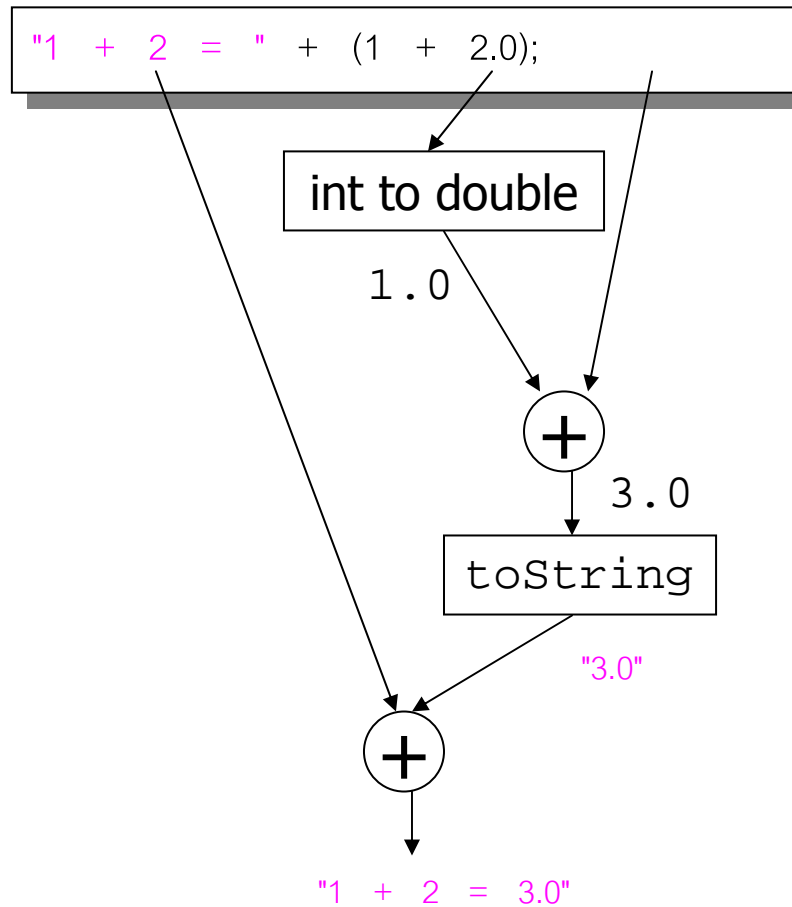
```
Approx. value of PI is 3.141592653589793
```

```
JLab>
```


ข้อควรระวัง : ลำดับการ +



ข้อควรระวัง : ลำดับการ +



String ไม่ใช่ประเภทข้อมูลพื้นฐานในจาวา

- **String** เป็น class มาตรฐานในจาวา
- 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'
```

บาง methods ของ **String**

- `length()` คืน `int` ซึ่งแทนความยาวของสายอักขระนี้
- `toUpperCase()` คืน `String` ซึ่งมีค่าเหมือนสายอักขระนี้ แต่ตัวอักษรอังกฤษทุกตัวเปลี่ยนเป็นตัวใหญ่
- `trim()` คือ `String` ซึ่งมีค่าเหมือนสายอักขระนี้ แต่ตัด spaces ที่อยู่ทางซ้ายและทางขวาออก

```
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"
```

บาง methods ของ **String** (ต่อ)

- substring(**int** b, **int** e)
 - รับ **int** สองจำนวน (สมมติให้ชื่อเป็น b และ e)
 - คืน **String** ที่เป็นสายอักขระย่อยเริ่มตั้งแต่ตัวที่ 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>
```

Methods ต่างๆ ของ **String**

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

charAt

contentEquals

equals

hashCode

lastIndexOf

regionMatches

replaceFirst

subSequence

toLowerCase

trim

compareTo

copyValueOf

getBytes

indexOf

length

replace

split

substring

toString

valueOf

concat

endsWith

getChars

intern

matches

replaceAll

startsWith

toCharArray

toUpperCase

Java API Help File

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

โปรแกรมแสดงชื่อเดือน

- Requirement

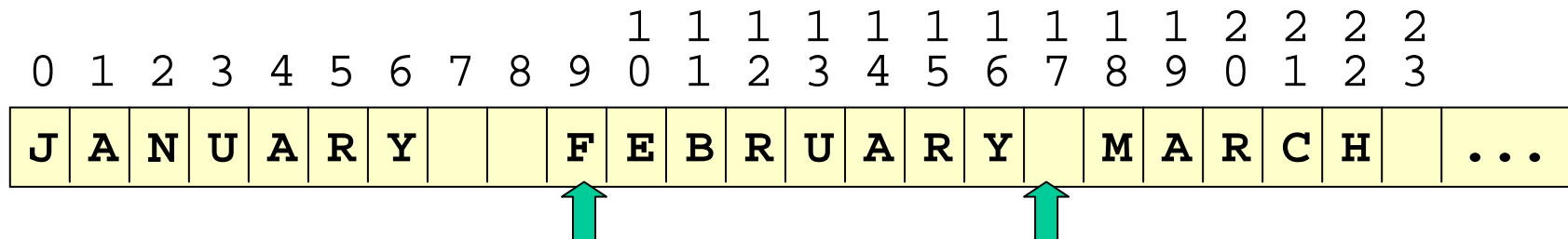
- แสดงชื่อเดือนจากหมายเลขเดือนที่ได้รับ

- Analysis

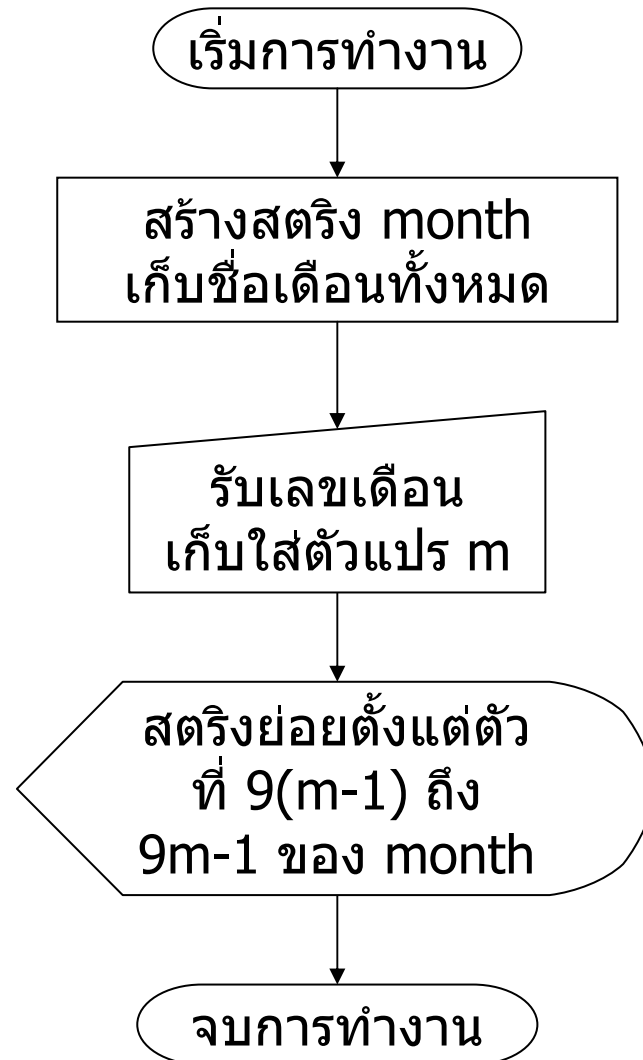
- รับหมายเลขของเดือนผ่านทางแป้นพิมพ์
- แสดงชื่อเดือนทางจอภาพ
- ข้อสังเกต : เดือนที่ชื่อยาวที่สุดคือ "SEPTEMBER"

- Design

- สร้างสายอักขระยาว $9 \times 12 = 108$ ตัว บรรจุชื่อเดือนเรียงต่อกันไป เดือนละ 9 ตัวอักษร



ผังงานการแสดงชื่อเดือน



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>
```

										1	1	1	1	1	1	1	1	1	1	2	2	2	2		
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	...

คำสั่ง console I/O พื้นฐานในจาวา

- แสดงออกจอภาพ
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

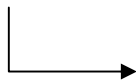
คำสั่ง console input ที่มีให้เฉพาะใน JLab

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

คำสั่ง console input ที่มีให้เฉพาะใน JLab

```
import jlab.JLabIO;
public class TestJLabIO {
    public static void main(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 ฟ้องเอง



```
JLab>java TestJLabIO
Name : Somchai
Age : 41
Income Tax : I wont 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

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



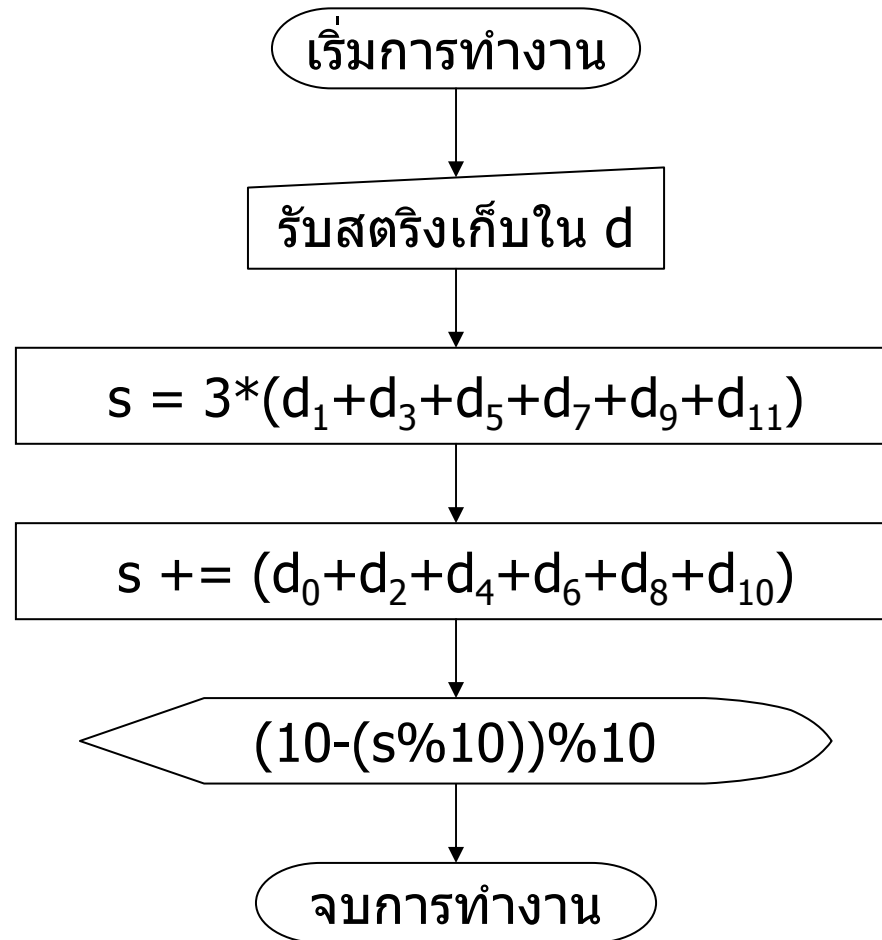
$$((7+1+3+5+7+9)+3(0+2+4+6+8+0)) = 92$$

$$(10-(92\%10))\%10 = 8$$

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);
    }
}
```