

2110211 : โครงสร้างข้อมูล

จาวาเบื้องต้น

สมชาย ประสิทธิ์จตุระกุล
ภาควิชาวิศวกรรมคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อ

- เรื่องทั่วไปของจาวา กับ 2110211
- Primitive data types
- Variables
- Operators
- if, if-else
- do-while, while, for

โปรแกรมแรก

```
// First Java Program

public class Hello {

    // main method
    public static void main(String[] args) {
        System.out.print("Hello ");
        System.out.println("2110211");
    }
} // end of class Hello
```

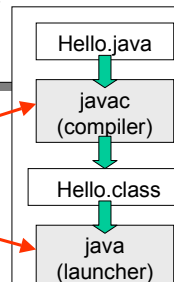
การแปลและสั่งทำงาน

```
// First Java Program

public class Hello {

    // main method
    public static void main(String[] args) {
        System.out.print("Hello ");
        System.out.println("2110211");
    }
} // end of class Hello
```

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



Source Codes, Byte Codes, Machine Codes

- Programmer เขียน source codes
- Java Compiler แปล source code เป็น byte codes
- Java Launcher นำ byte code ให้ Java Virtual Machine (JVM) ทำงาน
- JVM นำ byte codes มาเปลี่ยนเป็น machine codes ของเครื่องที่ใช้งาน

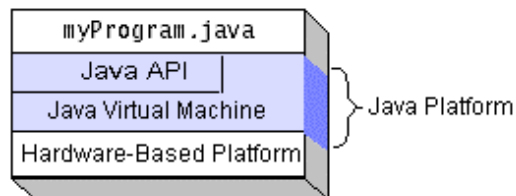
Java Programming Language

- The Java™ Language Specification : Second Ed. (James Gosling, Bill Joy, Guy Steele and Gilad Bracha)

```
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
```

Java Platform

- A *platform* is the hardware or software environment in which a program runs.
- Java Platform has two components :
 - The *Java Virtual Machine* (Java VM)
 - The *Java Application Programming Interface* (Java API)



โปรแกรมที่สอง

```
import java.util.Arrays;

public class A {

    public static void main(String[] args) {
        int[] data = new int[10];
        for (int i = 0; i < data.length; i++) {
            data[i] = (int) (100 * Math.random());
        }
        Arrays.sort(data, 0, data.length);
        for (int i = 0; i < data.length; i++) {
            System.out.print(data[i] + " ");
        }
        System.out.println();
    }
}
```

Java APIs

version	#packages	#classes
1.0	8	212
1.1	23	504
1.2	59	1520
1.3	76	~1800
1.4	135	~2800

Java Platform Editions

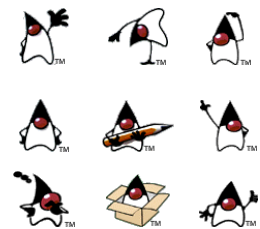
- Three editions of Java 2 Platform
 - Standard Edition (J2SE)
(client-side general-purpose applications)
 - Enterprise Edition (J2EE)
(multi-tier server-centric applications)
 - Micro Edition (J2ME)
(consumer and embedded devices)

JRE and JSDK

- JRE : the Java Runtime Environment
 - JVM : the Java virtual machine
 - API : the Java platform core classes
 - supporting files
- JSDK : the Java Software Development Kit
 - the JRE
 - development tools such as compilers and debuggers.

Java

- The name is not an acronym, but rather a reminder of that hot, aromatic stuff that many programmers like to drink lots of.
- A slang term for coffee that dates back to the days when the best brews came from Indonesia, not *Seattle*.



The Duke

ทำไมถึงใช้จาวา ?

- Object-oriented
- Cross platform : (Linux, Windows, Mac OS, ...)
- Internationalization (i18n)
- Huge class libraries (APIs)
- Popularity
- etc. (network-oriented, automatic garbage collector, strong type checking, secure, many free development tools, ...)

ปัญหาเริ่มต้นของผู้อยากเรียนรู้จาวา

- หนังสือที่เกี่ยวกับจาวา
 - มีมากมาย (www.amazon.com)
 - หนาๆ
 - แพงๆ
 - เก่าเร็ว
- ชุดพัฒนาโปรแกรมภาษาจาวา
 - ใหญ่ และใช้ยาก (Sun One Studio, JBuilder, ...)

จาวาสำหรับวิชา 2110211

- ตัวภาษา

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			

จาวาสำหรับวิชา 2110211

- รู้ object-oriented programming (บ้าง ไม่มาก)
- ใช้ Java APIs น้อยมาก (<5% ของที่มี)
 - ไม่มีการใช้ graphics, network, database, ...
- ใช้ Java 2 รุ่น 1.2 เป็นต้นไป (แนะนำ 1.4.1_01)
- ใช้ JLab เป็นซอฟต์แวร์ช่วยพัฒนาโปรแกรม

สถิติการทำโปรแกรม

- องค์ประกอบของคลาส
- main method

ประเภทข้อมูล

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

ประเภทจำนวนเต็ม (integer)

- จำนวนเต็ม (ไม่มีจุดทศนิยม)
เช่น 1023, 76289, 0, -10, +67392
- มีสี่ขนาด ขึ้นกับช่วงของจำนวนที่ใช้
 - `byte` (-128 ถึง 127)
 - `short` (-32768 ถึง 32767)
 - `int` (-2^{31} ถึง $2^{31} - 1$)
 - `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`

ประเภทอักขระ (character)

- ข้อมูลแบบ **char** คือตัวอักขระ**หนึ่ง**ตัว
 - จาวาใช้มาตรฐาน Unicode (char หนึ่งตัวใช้สองไบต์)
- อักขระ (character) คือ
 - ตัวอักษร ตัวเลขโดด สัญลักษณ์พิเศษ เช่น 'a', 'b', 'A', 'B', 'Ω', 'Ж', 'ก', 'ข', '1', '2', '๑', '๒', '#', '\$', '฿', '๘'
 - รหัสควบคุมอุปกรณ์
 - '\n' ตัวขึ้นบรรทัดใหม่
 - '\t' ตัว tab
 - '\\' สัญลักษณ์ \ (backslash)
 - '\'' สัญลักษณ์ ' (single quote)
 - '\"' สัญลักษณ์ " (double quote)

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**

การประกาศตัวแปร

```
int counter = 0;
double radius = 5.225;
boolean success = false;
char korKai = 'ก';
String name = "somchai";
```

ประเภทข้อมูล ชื่อตัวแปร ค่าเริ่มต้นของข้อมูลที่เก็บในตัวแปร

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;
```

นิยามประเภทข้อมูลที่ไม่ว่าง

```
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
- ต้องไม่ซ้ำกับคำสงวนของภาษาจาวา
- ต้องไม่ซ้ำกับชื่ออื่นๆ ของโปรแกรม (เช่น ชื่อ class)

ตัวอย่างถูก

```
int17 butterCup Public int2String day_of_week
```

ตัวอย่างผิด

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

ธรรมเนียมนิยมในการตั้งชื่อ

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

```
numStudents dayOfWeek imageBuffer
```

```
tvChannel customerID initialVelocity
```

เรื่องพื้นฐาน

- การให้ค่าตัวแปร (assignment) ใช้เครื่องหมาย =
- ตัวปฏิบัติการพื้นฐาน + - * / %

ตัวปฏิบัติการพื้นฐาน : จำนวนเต็มและจริง (ต่อ)

- การเพิ่มค่าทีละหนึ่ง มีสองแบบ
 - a++ นำค่าของ a ไปใช้ แล้วค่อยเพิ่มค่าของ a ขึ้นอีกหนึ่ง
 - ++a เพิ่มค่าของ a ขึ้นอีกหนึ่ง แล้วค่อยนำค่าไปใช้

	int i	int j	int k
int i=1, j=2, k;	1	2	?
i++;	2	2	?
++j;	2	3	?
j = i++;	3	2	?
k = ++j + i++;	4	3	6
k = j++ + ++i;	5	4	8

ทำเป็นลำดับจากบนลงล่าง

การเปลี่ยนแปลงค่าของตัวแปร

ตัวปฏิบัติการพื้นฐาน : จำนวนเต็มและจริง (ต่อ)

- การลดค่าทีละหนึ่ง มีสองแบบ
 - a-- นำค่าของ a ไปใช้ แล้วค่อยลดค่าของ a ลงหนึ่ง
 - --a ลดค่าของ a ลงหนึ่ง แล้วค่อยนำค่าไปใช้

	int i	int j	int k
int i=10, j=4, k;	10	4	?
i--;	9	4	?
--j;	9	3	?
j = --i;	8	8	?
k = --j - --i;	7	7	0
k = j-- - --i;	6	6	1

ทำเป็นลำดับจากบนลงล่าง

การเปลี่ยนแปลงค่าของตัวแปร

ตัวอย่างงง

```
double x=10.0, y=20.0, z;
```

z = x++ + ++y;	}	compiler ไม่งง
z = x-- - --y;		
z = x++-++y;		
z = x--+-y;		
z = x+++++y;	}	compiler งง
z = x-----y;		
(x++)++;		ทำไม่ได้

การเขียนค่าคงตัวที่เป็นจำนวนเต็ม

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

```
int base10 = 2545;
int base8 = 04761;
int base16 = 0x9F1;
```

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


การเขียนค่าคงตัวที่เป็นจำนวนจริง

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

ใช้ตัว e เล็กก็ได้

เลขชี้กำลังต้องเป็นจำนวนเต็ม

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 ?

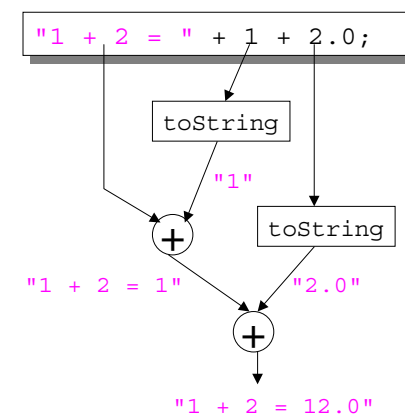
ตัวปฏิบัติการ + กับ String

- **String** มีตัวปฏิบัติการตัวเดียวคือ +
- + คือการต่อสายอักขระสองสายเข้าด้วยกัน
- เมื่อข้อมูลแบบ **String** + กับข้อมูลใด ข้อมูลนั้นจะถูกเปลี่ยนเป็น **String** ก่อน แล้วจึง +

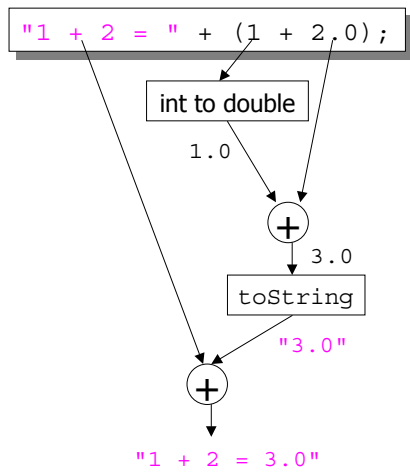
```
public class StringConcatenation {
    public static void main(String [] args) {
        String s1 = "Approx. value of PI";
        String s2 = "is";
        System.out.println(s1 + s2);
        System.out.println(s1 + " " + s2);
        System.out.println(s1 + " " + s2 + " " + Math.PI);
    }
}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

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

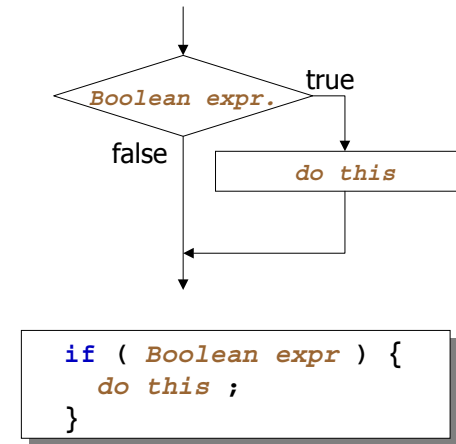
ข้อควรระวัง : ลำดับการ +



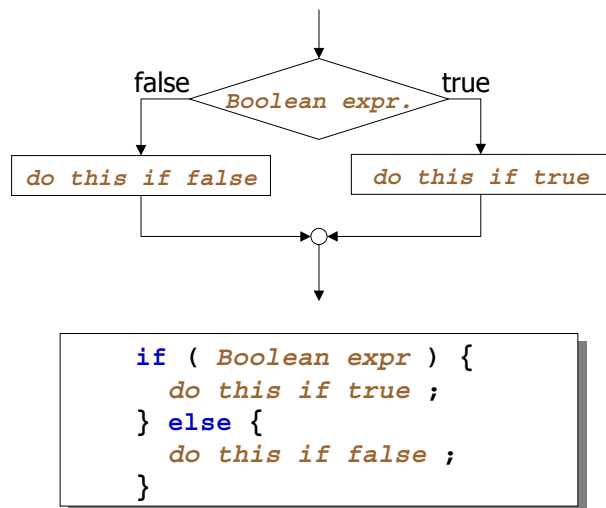
ข้อควรระวัง : ลำดับการ +



Selection : if statement



Selection : if-else statement



Relational Operators

- ตัวเปรียบเทียบข้อมูลประเภทพื้นฐาน ได้แก่

<	(น้อยกว่า)	>	(มากกว่า)
<=	(น้อยกว่าหรือเท่ากับ)	>=	(มากกว่าหรือเท่ากับ)
==	(เท่ากับ)	!=	(ไม่เท่ากับ)
- ได้ผลลัพธ์เป็นประเภท **boolean**

((n % 2) == 1) ได้ค่าเป็น

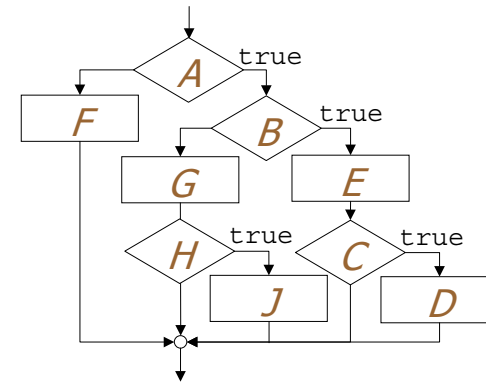
- true ถ้า n เป็นจำนวนคี่
- false ถ้า n เป็นจำนวนคู่

n = 1 ไม่เหมือน n == 1

Boolean Operators

- ใช้เครื่องหมาย && แทน and
- ใช้เครื่องหมาย || แทน or
- ใช้เครื่องหมาย ! แทน not

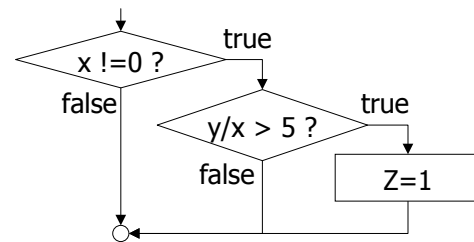
Nested If



```

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

Short-circuiting



```

if ( x != 0 ) {
    if ( y/x > 5 ) {
        z = 1 ;
    }
}
    
```

```

if ( x != 0 && y/x > 5 ) {
    z = 1 ;
}
    
```

```

if ( y/x > 5 && x != 0 ) {
    z = 1 ;
}
    
```

ถ้า $x == 0$ จะไม่มีการคำนวณ y/x

คำนวณ y/x ก่อนทดสอบ $x != 0$

Short-circuiting (ไม่คิดหมด พยายามหาทางลัด)

- กรณี && จะเลิกประเมิน expr ทางขวาถัดไป ถ้าพบ expr ที่เป็น false
- กรณี || จะเลิกประเมิน expr ทางขวาถัดไป ถ้าพบ 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

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

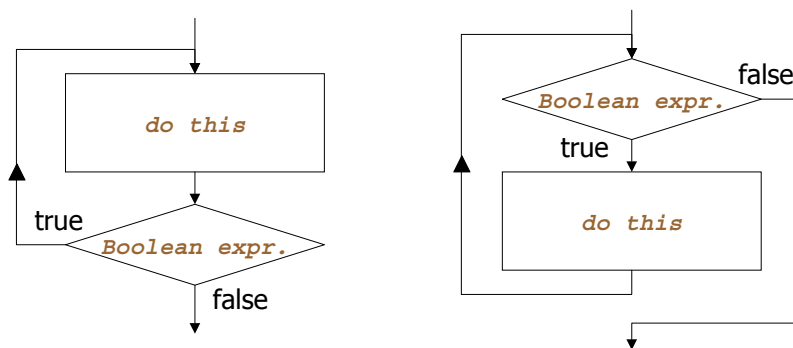
อย่าใช้ == < > เพื่อเปรียบเทียบ String

- **String** ไม่ใช่ประเภทพื้นฐาน
- 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)` $\begin{cases} < 0 & \text{ถ้า } s1 \text{ มาก่อน } s2 \text{ ตามลำดับตัวอักษร} \\ = 0 & \text{ถ้า } s1 \text{ เหมือนกับ } s2 \\ > 0 & \text{ถ้า } s1 \text{ มาหลัง } s2 \text{ ตามลำดับตัวอักษร} \end{cases}$

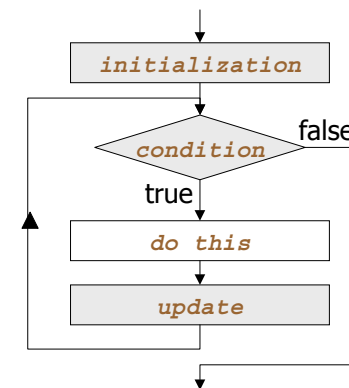
วงวนแบบ do-while และ while



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

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

วงวนแบบ for



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

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 เพื่อออกจากวงวน

```
for ( int j=0, i=0; j<n; ++j ) {  
    while ( i < n ) {  
        ...  
        ... break;  
    }  
    ... break;  
}
```

คำสั่ง continue เพื่อทำรอบถัดไป

```
int i = 0;  
  
while ( i < 7 ) {  
    i++;  
    System.out.println(i);  
    if ( i > 3 ) continue;  
    System.out.println(i + 100);  
}
```

โปรแกรมหารากที่สอง

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

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)

$a = 4.0$
 $x_0 = 1.0$
 $x_1 = 2.5$
 $x_2 = 2.05$
 $x_3 = 2.000609756097561$
 $x_4 = 2.0000000929222947$
 $x_5 = 2.0000000000000002$
 $x_6 = 2.0$

$a = 100.0$
 $x_0 = 1.0$
 $x_1 = 50.5$
 $x_2 = 26.24009900990099$
 $x_3 = 15.025530119986813$
 $x_4 = 10.840434673026925$
 $x_5 = 10.032578510960604$
 $x_6 = 10.000052895642693$
 $x_7 = 10.000000000139897$
 $x_8 = 10.0$

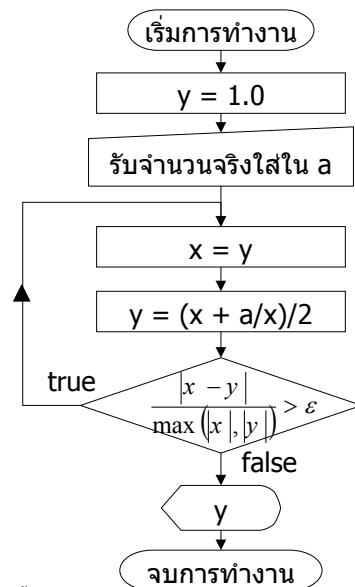
แล้วจะประมาณค่าไปสักกี่รอบ ?

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

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

(ให้ ϵ มีค่าเท่ากับ 10^{-14})

ผังงานการหารากที่สอง



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);
    }
}
    
```

หมุนในวงวน 339 รอบ

```

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