

จาวา - เบื้องต้น

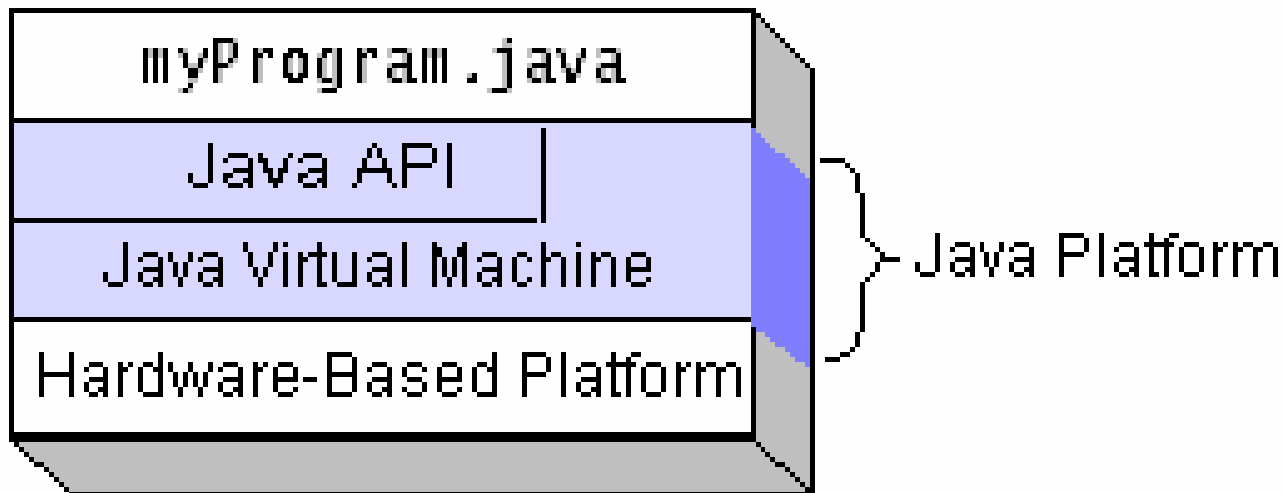
สมชาย ประสิทธิ์จตุระกุล

เนื้อหา

- ระบบบจาวา
- โครงของโปรแกรม
- ประเภทข้อมูล
- ข้อมูลพื้นฐานชนิดต่างๆ
- ตัวแปร
- ตัวดำเนินการต่างๆ
- การเปลี่ยนประเภทข้อมูล
- โครงสร้างการทำงานแบบต่างๆ
- อারেย์

Java Technology

- Java Programming Language
- Java Platform
 - Java Virtual Machine
 - Java API



Solaris, Linux, Windows, Mac OS X, HP-UX, IBM-AIX,
OS/390, PDA, Java-enabled cell phones, GameBoy Advance, ...

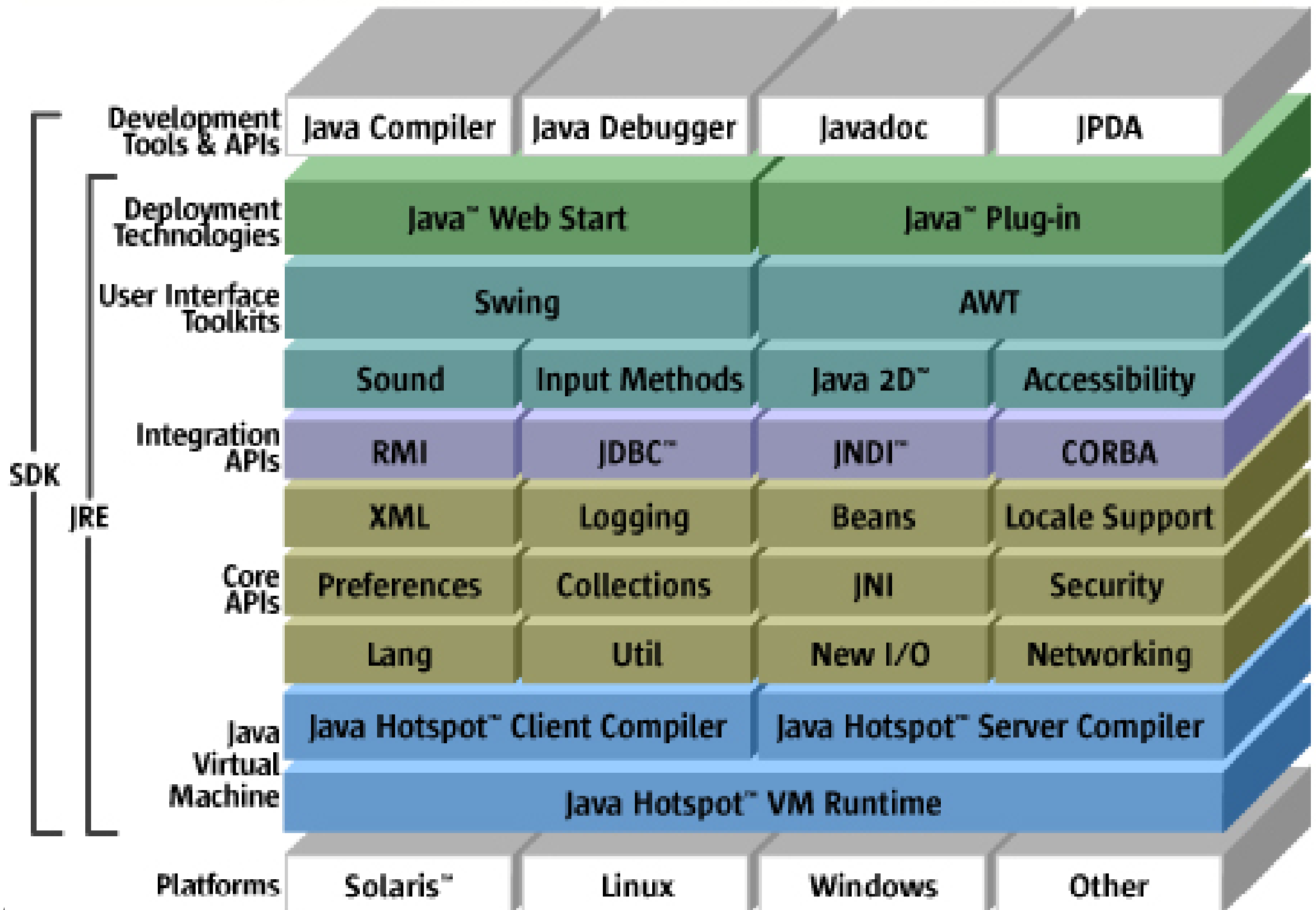
Language Features

- 1995 (1.0) : first public release
 - A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language.
- 1997 (1.1) : nested classes
- 2001 (1.4) : assert
- 2004 (1.5 หรือ Java 5) :
 - generics, enum, enhanced for, autoboxing/unboxing, varargs, static import, metadata.

Java Platforms

- Standard Edition (J2SE)
 - client-side general-purpose applications
- Enterprise Edition (J2EE)
 - multi-tier server-centric applications
- Micro Edition (J2ME)
 - consumer and embedded devices

Java™ 2 Platform, Standard Edition v 1.4



Java API

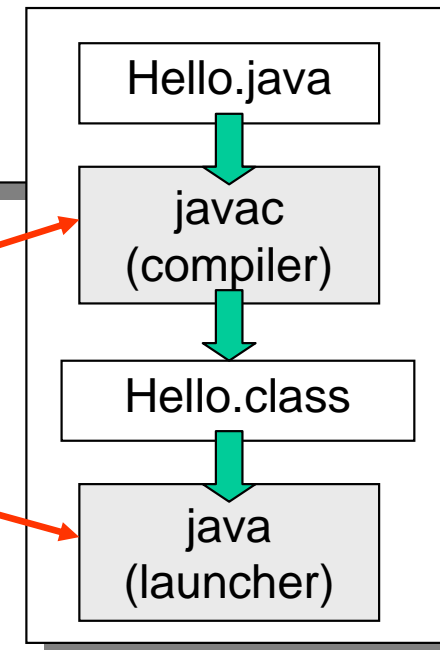
	version	#packages	#classes
	1.0	8	212
	1.1	23	504
Java 2	1.2	59	1520
	1.3	76	~1800
	1.4	135	~2800
	1.5	164	~3300

การแปลและสั่งทำงาน

```
/* First Java Program
 */
package com.somchai;

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

```
C:\>javac Hello.java
C:\>java Hello
Hello World
C:\>
```



Comments

```
/* First Java Program
 */
package com.somchai;

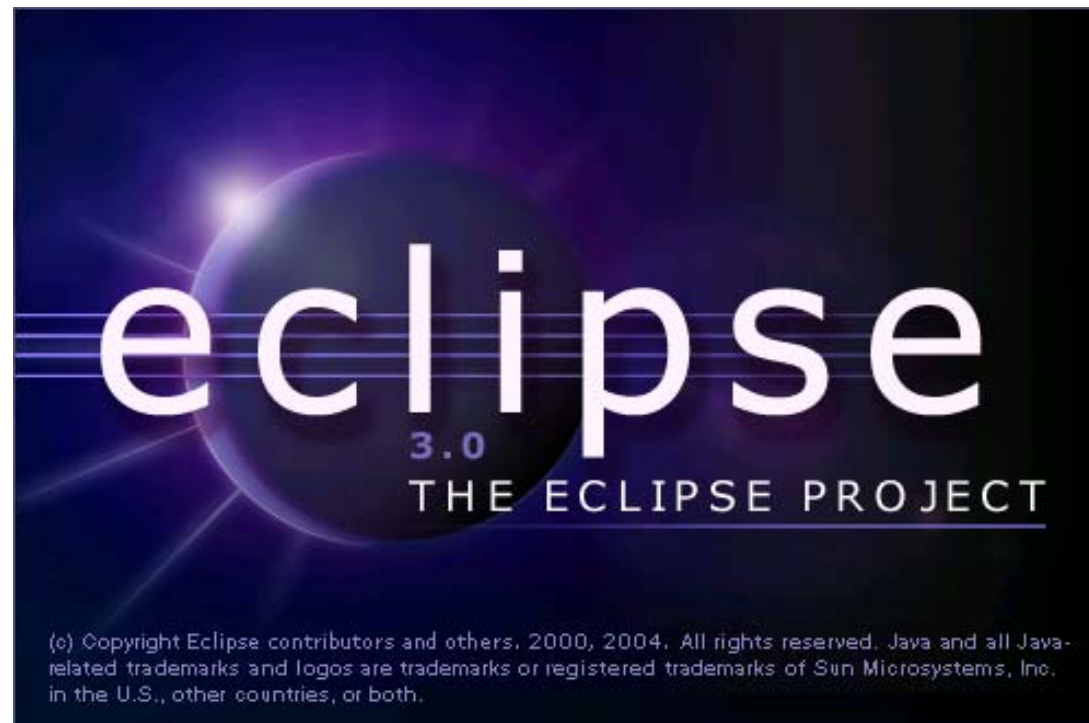
public class Hello {
    public static void main(String[] args) {
        System.out.print("Hello ");
        System.out.println("World");
    }
} // end of class Hello
```

comment

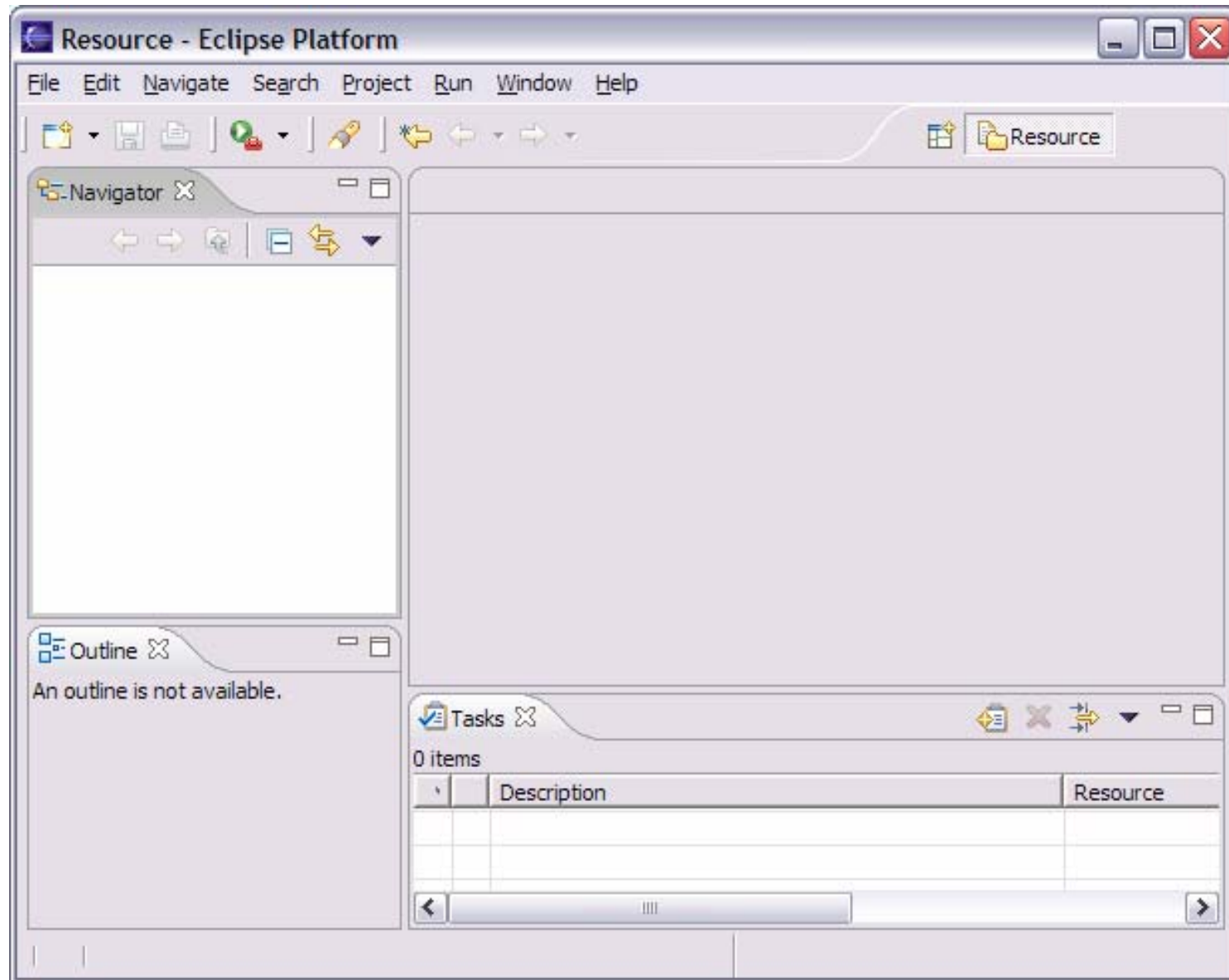


www.eclipse.org

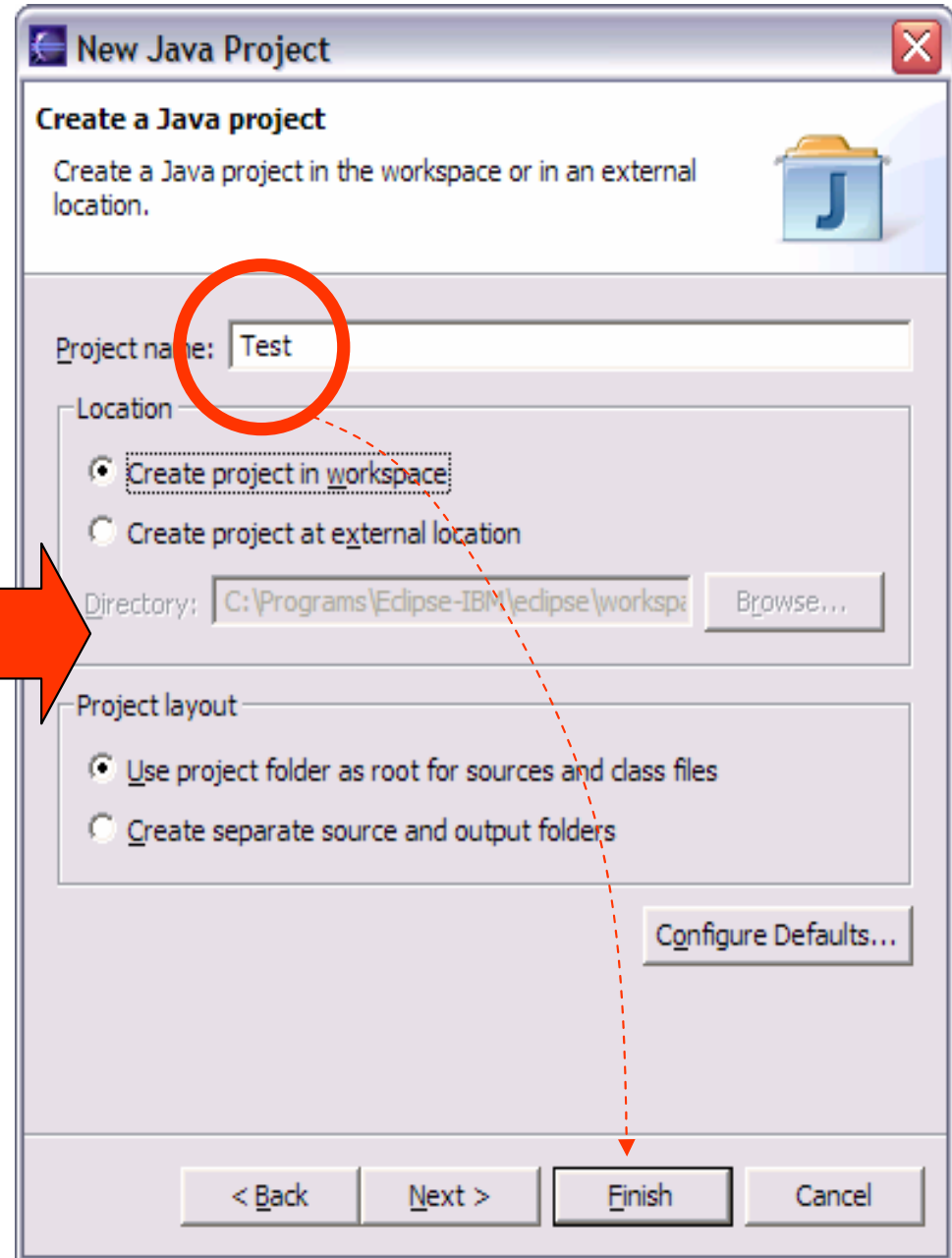
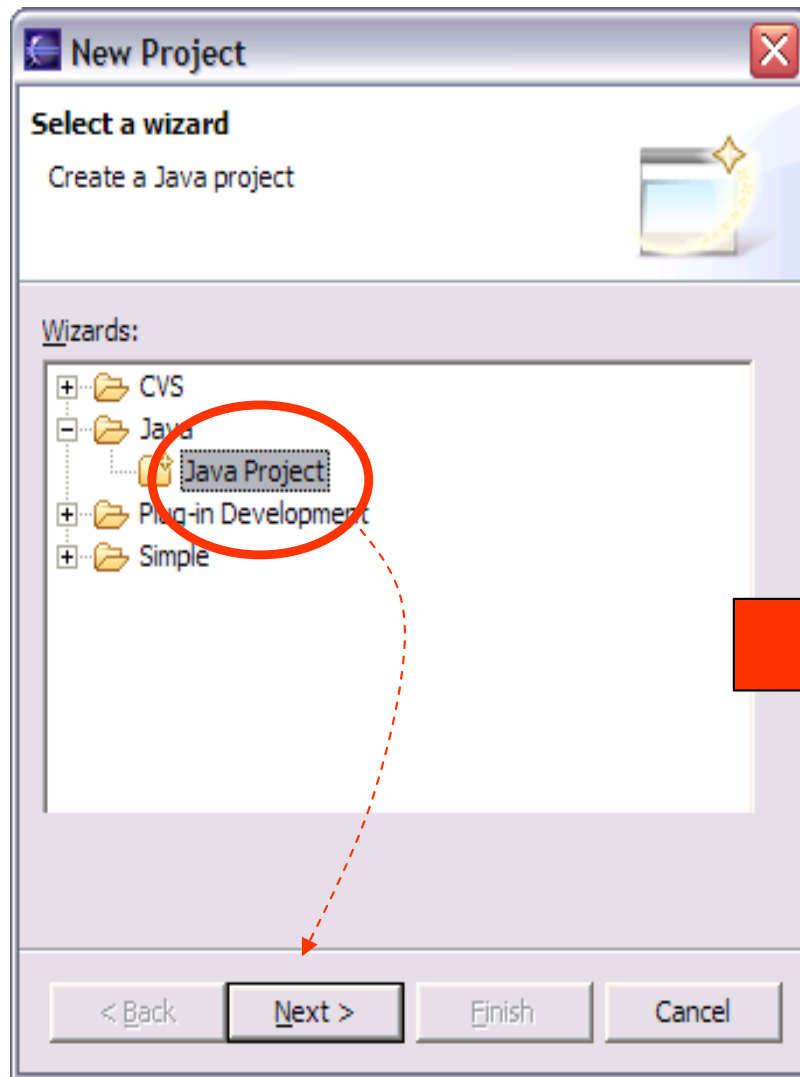
- ติดตั้ง j2sdk (<http://java.sun.com/j2se>)
- ติดตั้ง eclipse (<http://www.eclipse.org>)
 - download “eclipse-SDK-?????.zip”
 - unzip และ execute “eclipse.exe”



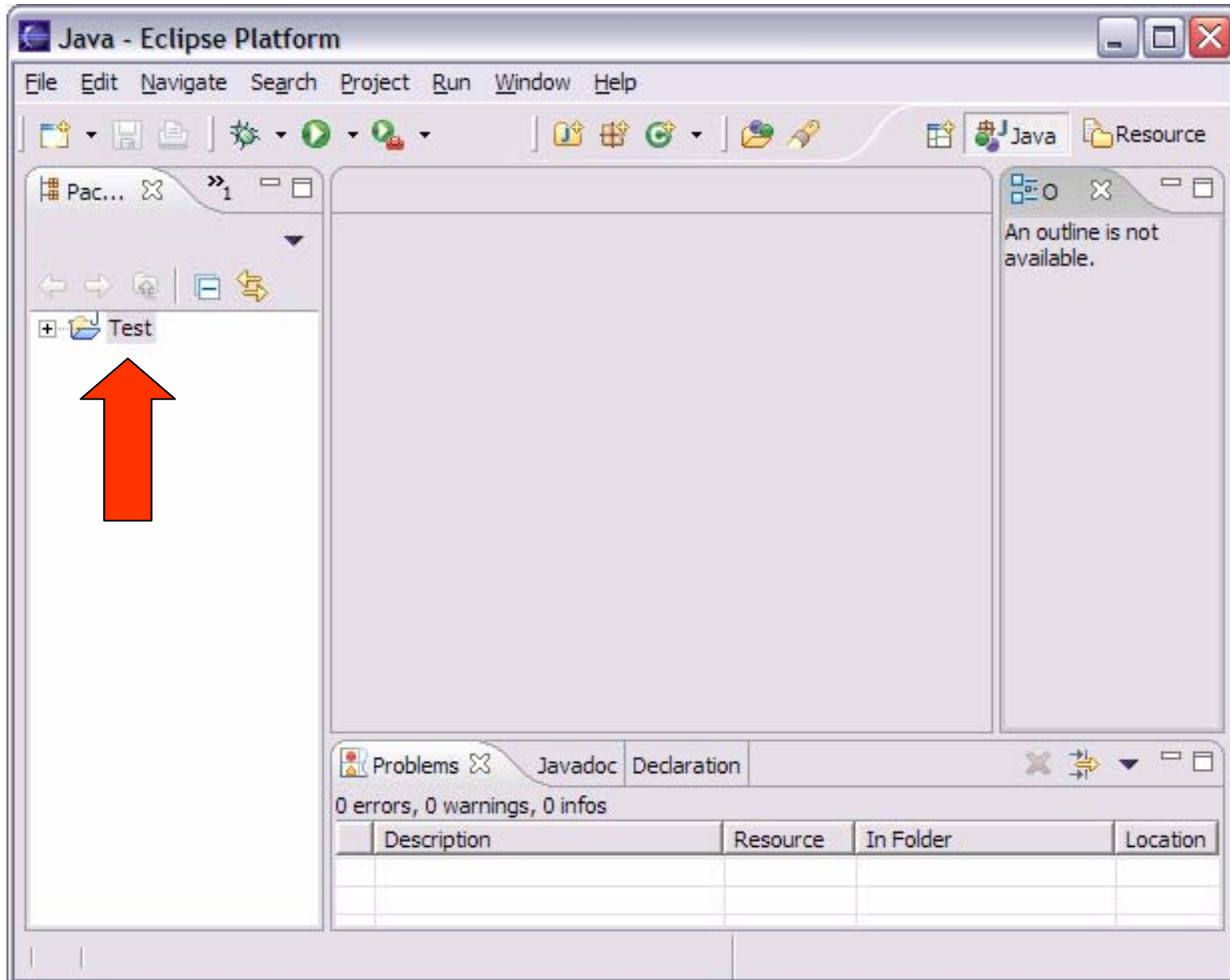
Eclipse Workbench



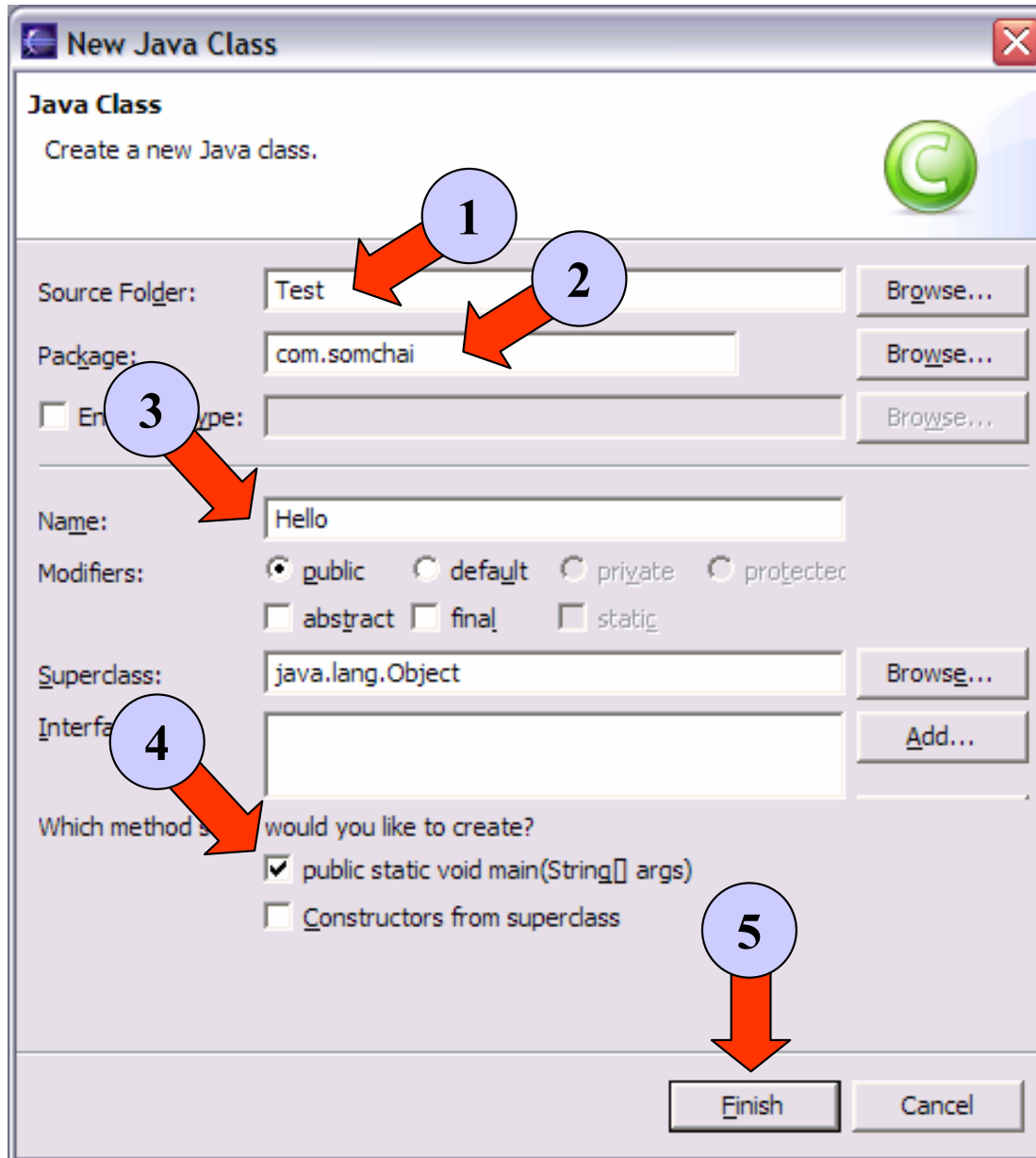
File -> New -> Project...



Java Perspective



File -> New -> Class



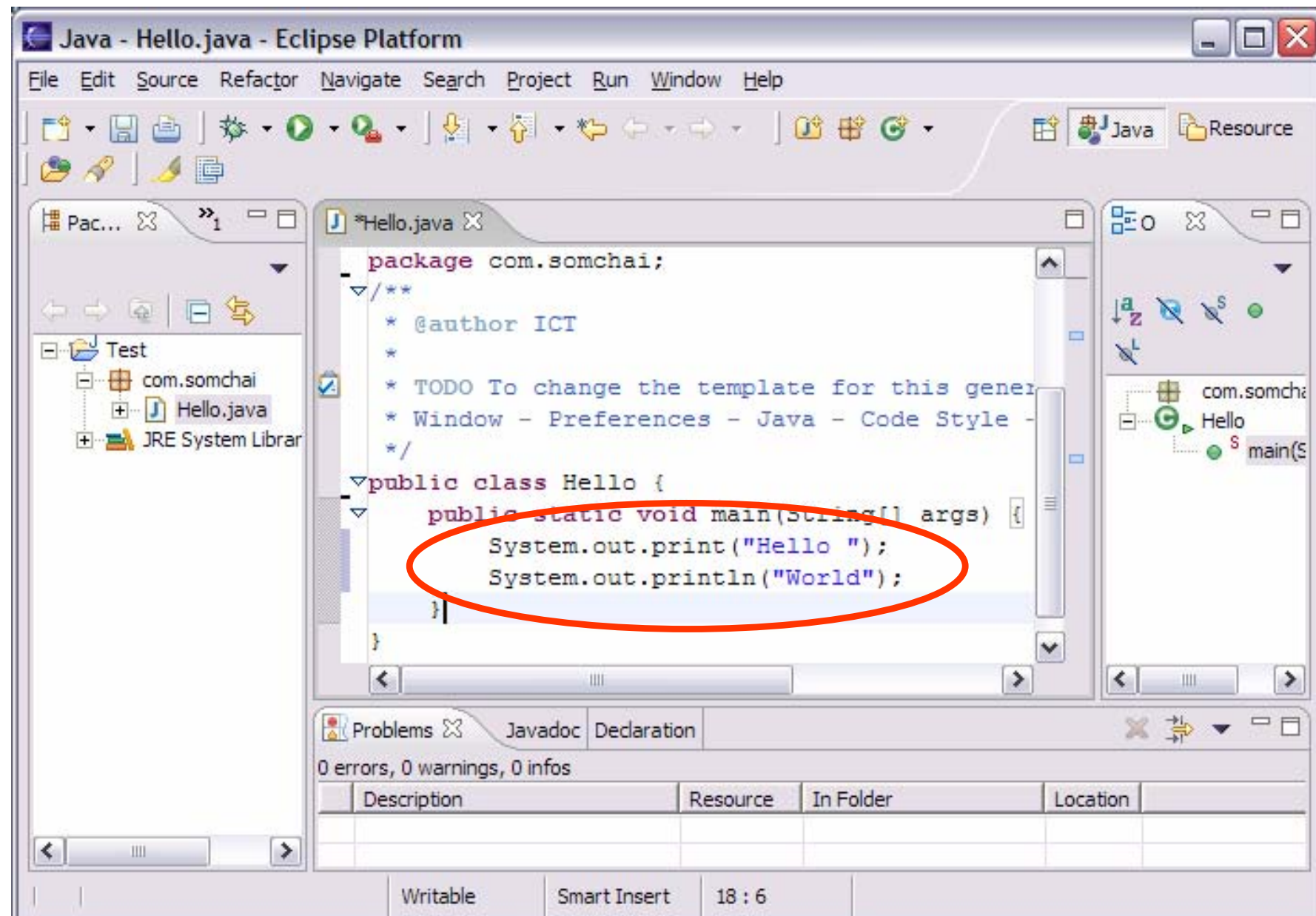
ตั้งชื่อคลาส Hello

อยู่ในแพ็คเกจ com.somchai

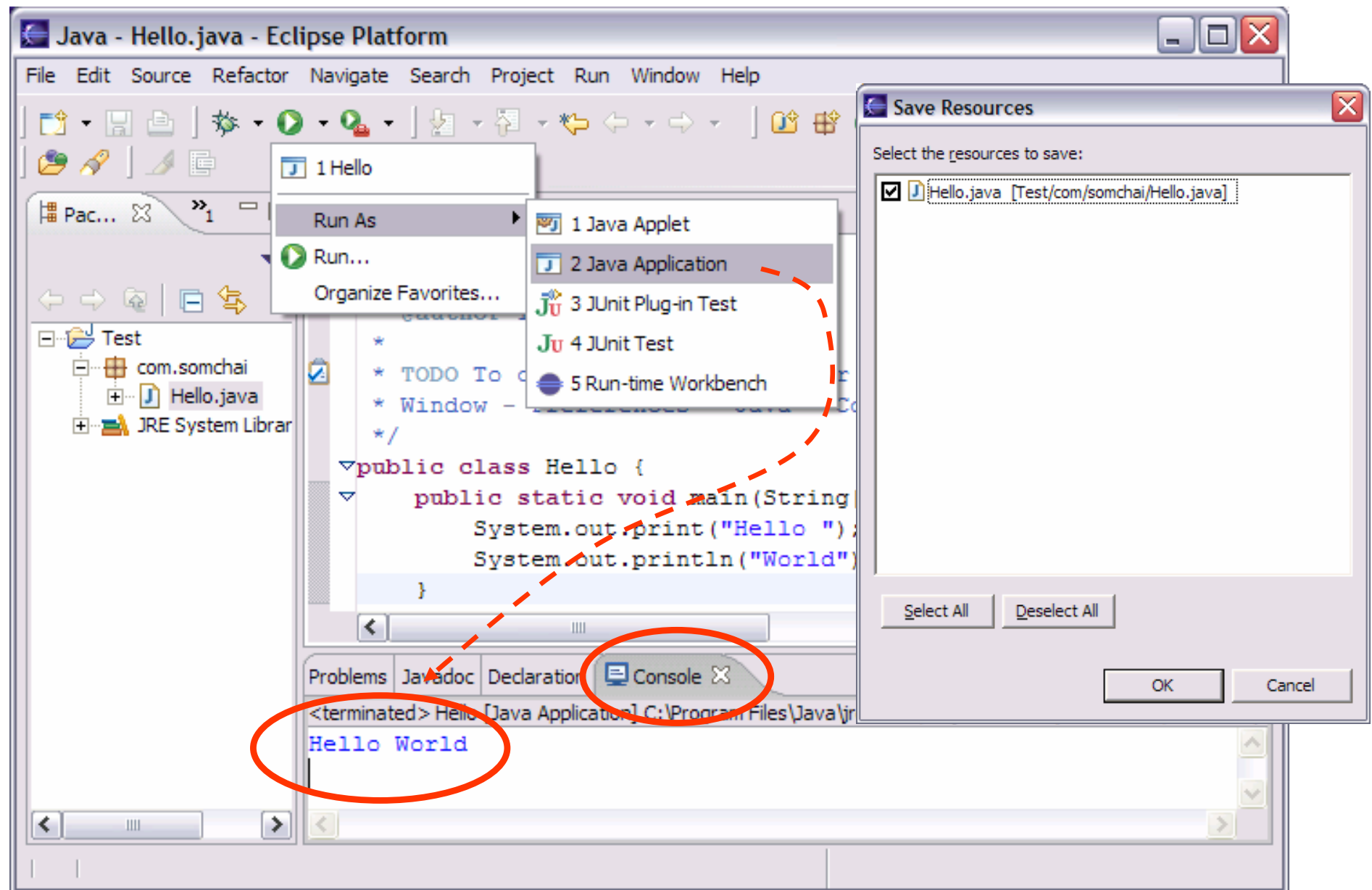
คลาสใหม่นี้มีชื่อเต็ม

com.somchai.HelloWorld

Editor



Run -> Run As -> Java Application



ประเภทข้อมูล

- ประเภทพื้นฐาน

- มี 8 ประเภท

- จำนวนเต็ม : `byte` , `short` , `int` , `long`

- จำนวนจริง : `float` , `double`

- อักขระ : `char`

- ตรรกะ : `boolean`

- ออกแบบเพิ่มเติมไม่ได้

- ประเภท class

- มีแบบมาตรฐานหลายพันแบบ

- เช่น `String` , `Text` , `Rectangle` , ...

- ออกแบบเองเพิ่มเติมได้

Primitive Data Types

- จำนวนเต็มมีสี่ขนาด

- **byte** 8 บิต (-2^7 ถึง $2^7 - 1$)
- **short** 16 บิต (-2^{15} ถึง $2^{15} - 1$)
- **int** 32 บิต (-2^{31} ถึง $2^{31} - 1$)
- **long** 64 บิต (-2^{63} ถึง $2^{63} - 1$)

- จำนวนจริงมีสองขนาด

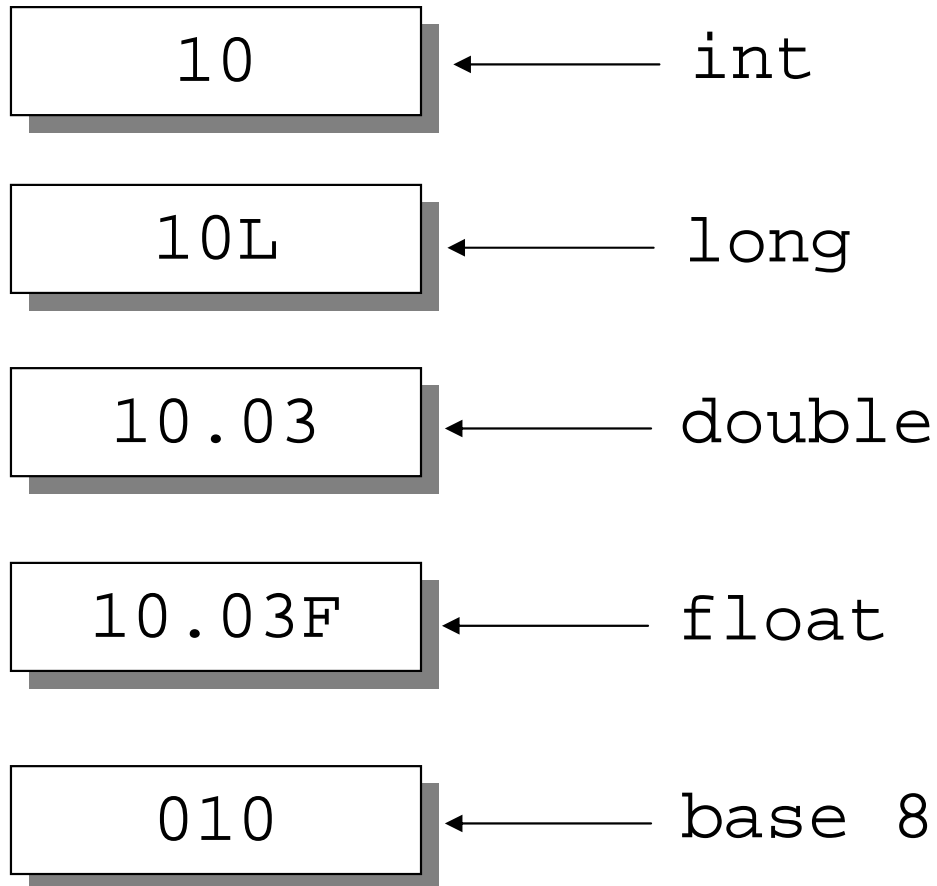
- **float** เก็บประมาณ 6-9 ตำแหน่งหลังจุดทศนิยมในช่วง (-3.4×10^{38} ถึง -1.4×10^{-45} ศูนย์ และ 1.4×10^{-45} ถึง 3.4×10^{38})
- **double** เก็บประมาณ 15-17 ตำแหน่งหลังจุดทศนิยมในช่วง (-1.8×10^{308} ถึง -4.9×10^{-324} ศูนย์ และ 4.9×10^{-324} ถึง 1.8×10^{308})

โดยทั่วไปใช้ int กับ double

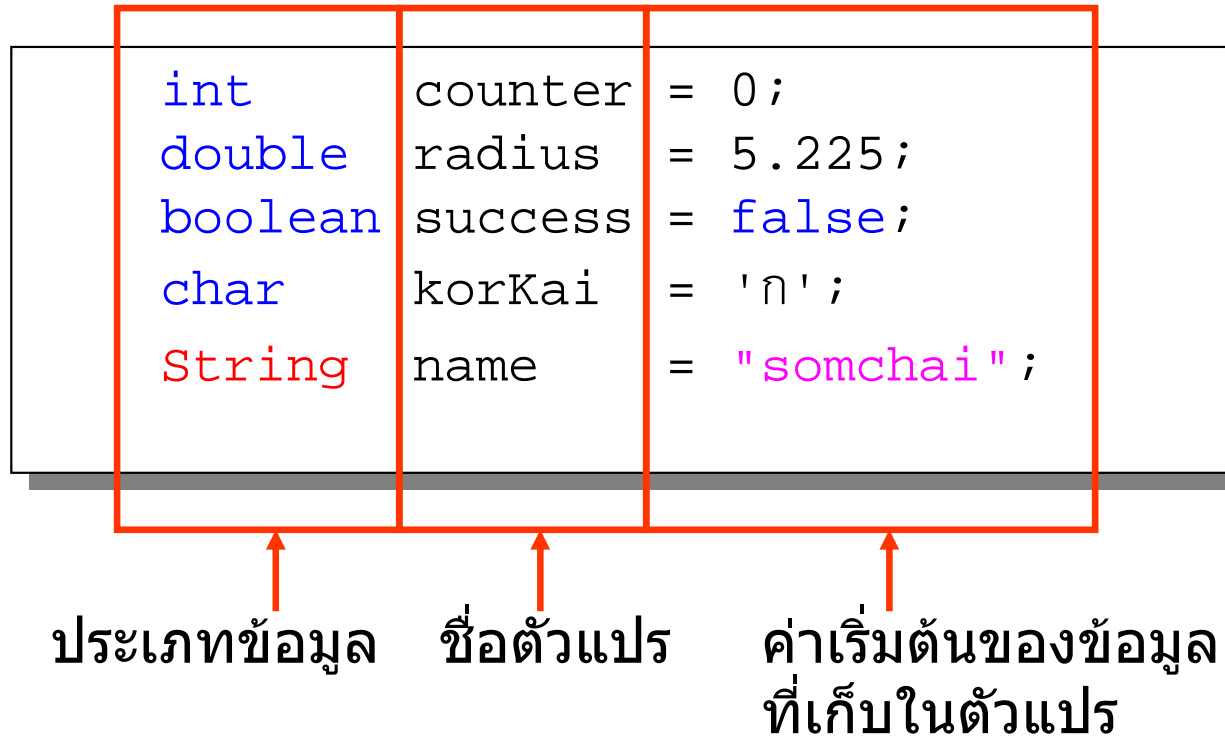
Primitive Data Types

- **boolean**
 - มีได้แค่สองค่าคือ `true` และ `false` เท่านั้น
 - 1 ไม่ได้แทน `true` 0 ไม่ได้แทน `false`
- **char** ใช้ Unicode (16 บิตต่อ 1 ตัว)
 - `'A'`, `'B'`, `'ก'`, `'β'`, `'\n'`
- **String**
 - ไม่ใช่ primitive data type
 - `"hello"`, `""`, `"hello\nworld"`

การเขียนค่าคงตัวที่ของจำนวน



การประกาศตัวแปร



ประกาศบรรทัดละหลายตัว หรือจะไม่ใส่ค่าเริ่มต้นก็ได้

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

ตัวอย่างการประกาศตัวแปรที่ผิด

```
int counter
```

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

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

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

กฎการตั้งชื่อ

- ชื่อประกอบด้วยตัวอักษร ตัวเลข ตัว \$ หรือ _ ก็ได้
- ชื่อห้ามขึ้นต้นด้วยตัวเลข
- ชื่อยาวๆ ได้ไม่เป็นไร
- ตัวอักษรตัวใหญ่ **ไม่เหมือน** ตัวเล็ก
- ต้องไม่ซ้ำกับคำสงวนของภาษาจาวา

ตัวอย่างถูก

```
int17 butterCup Public int2String day_of_week
```

ตัวอย่างผิด

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

คำสงวนในภาษาจาวา

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	enum		

ห้ามนำคำสงวนมาตั้งเป็นชื่อตัวแปร

Operators

- + - * / ++ --
- % เศษของการหาร
- == เท่ากัน ?
- != ไม่เท่ากัน ?
- < <= > >=
- ! NOT
- && AND
- || OR
- + เมื่อใช้กับ String จะหมายถึง concatenation

```
String s = "answer = " + 1 + 2 + 3; //answer123
```

การเปลี่ยนประเภทข้อมูล

- Widening

- เปลี่ยนจากประเภทข้อมูลที่มีช่วงแคบกว่าไปยังประเภทข้อมูลที่มีช่วงกว้างกว่า

```
int i = 123;  
long w = i;  
double d = 2;  
double x = i / d;
```

- Narrowing

- กลับกัน, ข้อมูลอาจหาย
- ถ้าต้องการ narrow ต้องทำ casting

```
float f = (float) 4.2;  
int i = (int) (f / 2);
```

```
byte < short < int < long < float < double
```

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 ?

Statements and Blocks

```
int x = 0, y = 0;  
x = x + 1; x = 2 * x;  
y =  
x;
```

```
int y = 0;  
{  
    int x = 0;  
    x = x + 1;  
    x = 2 * x;  
    y = x; // can use y  
}  
x++; // compile-error
```

Selection : if, if-else

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

```
if (Boolean Expression) {  
    statement(s) ;  
}
```

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

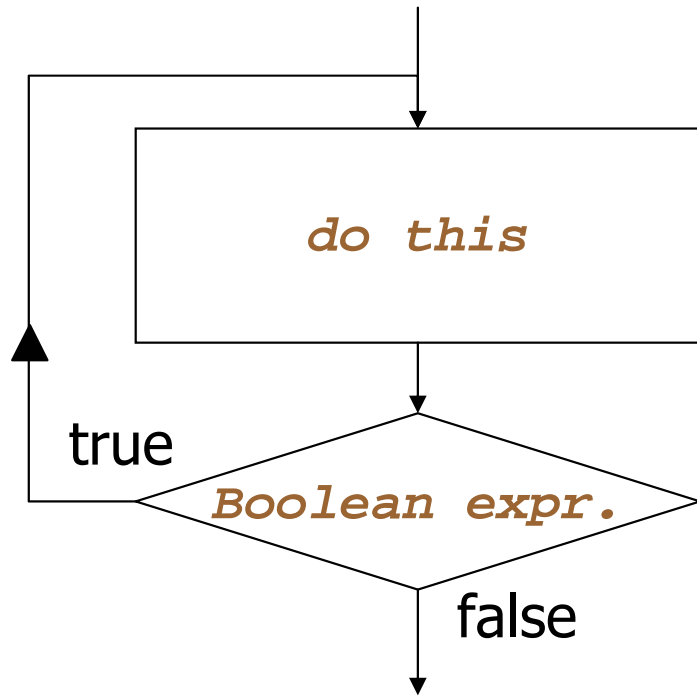
```
if (Boolean Expression) {  
    statement(s) ;  
} else {  
    statement(s) ;  
}
```

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

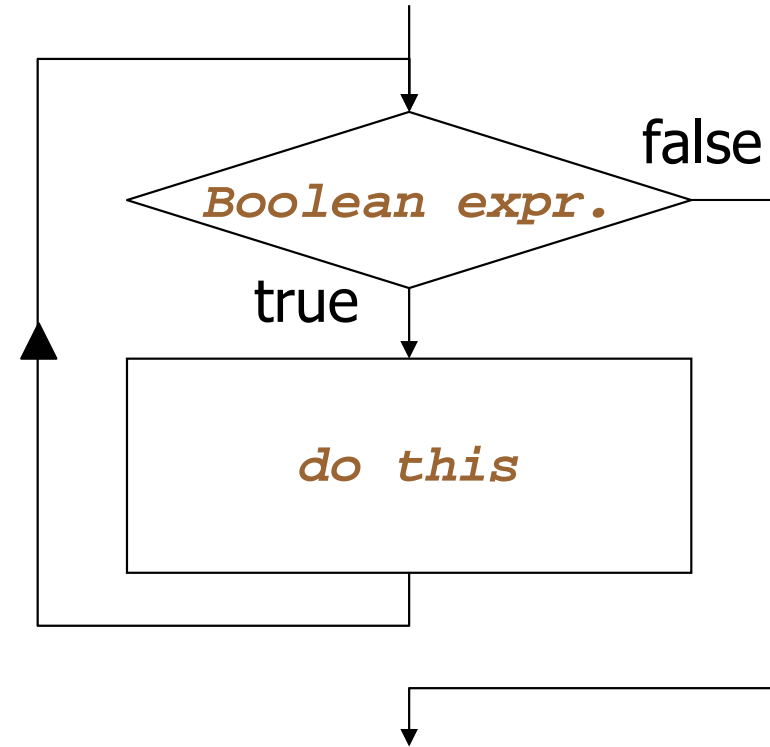
Selection : switch-case

```
int option = 5;
...
switch (option) {
    case 1 :
    case 21 :
        System.out.println("1 or 21");
        break;
    case 3 :
        System.out.println("3");
        break;
    default :
        System.out.println("not 1, 21, and 3");
        break;
}
```

Loops : do-while, while

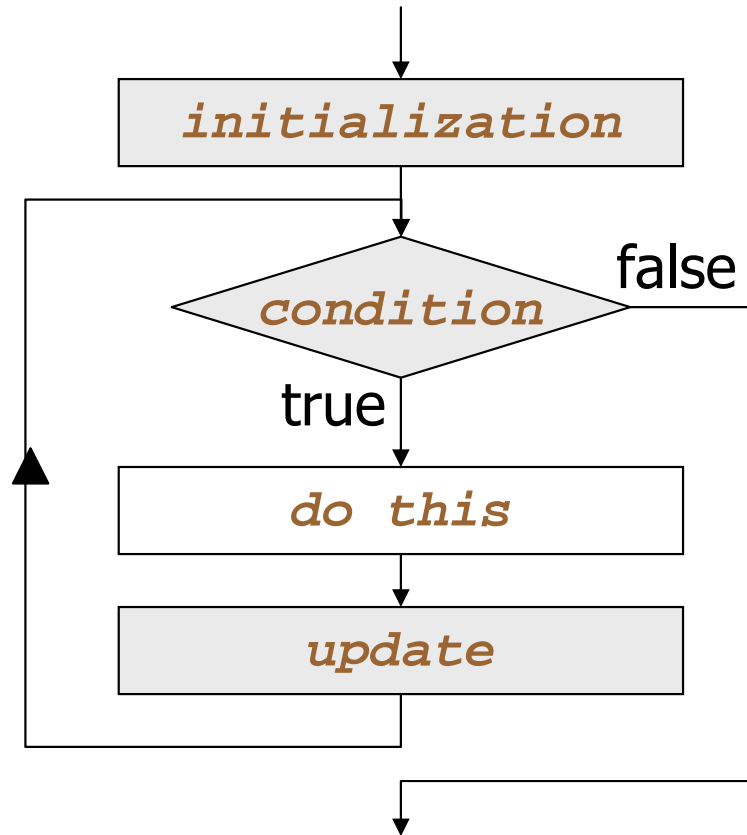


```
do {  
    do this  
} while ( Boolean expr );
```



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

Loops : for



```
i = 0;  
while( i < 30 ) {  
    System.out.println(i);  
    i++;  
}
```

```
for( i=0; i < 30; i++ ) {  
    System.out.println(i) )  
}
```

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


Variable Declaration in for

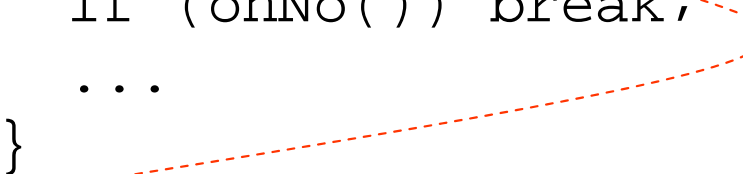
```
{  
  int i;  
  for( i = 0; i < 30; i++ ) {  
    ...  
  }  
}
```

```
for( int i = 0; i < 30; i++ ) {  
  ...  
}
```

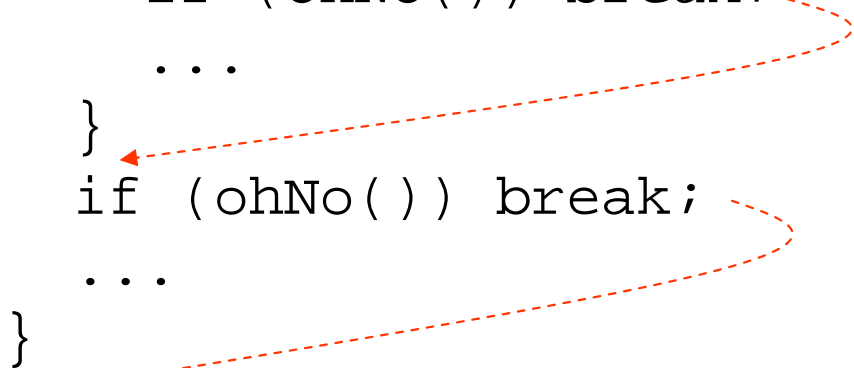
declaration statement

break

```
while (! success()) {  
    ...  
    if (ohNo()) break;  
    ...  
}
```

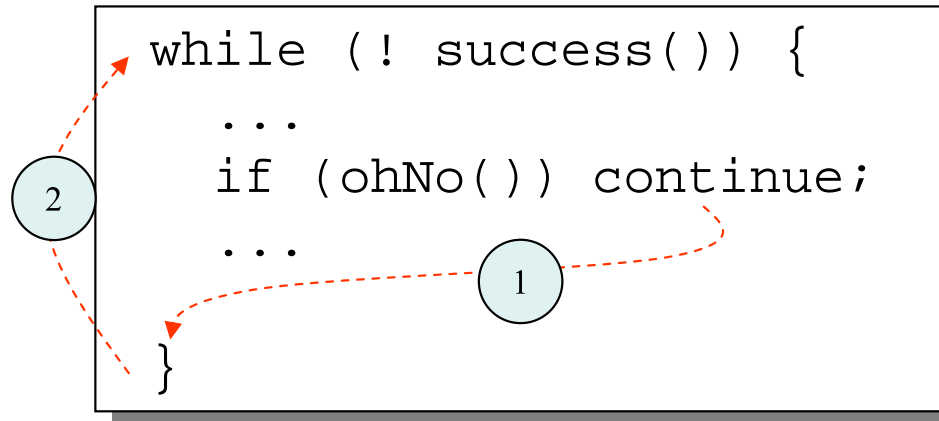


```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < n; j++) {  
        if (ohNo()) break;  
        ...  
    }  
    if (ohNo()) break;  
    ...  
}
```

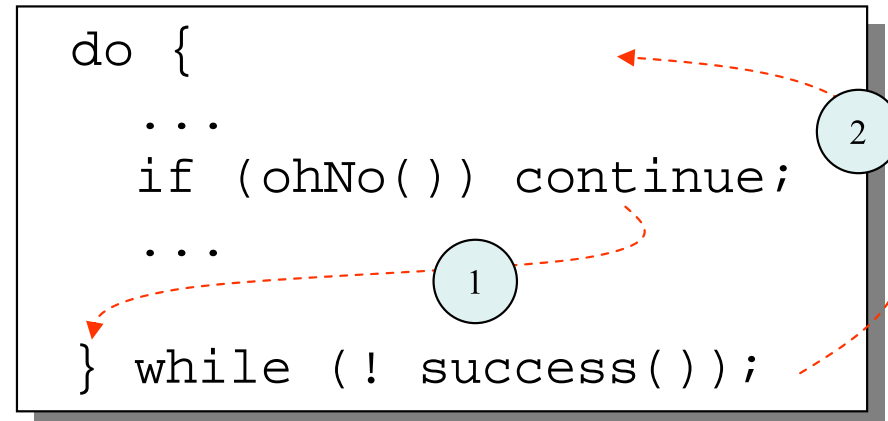


continue

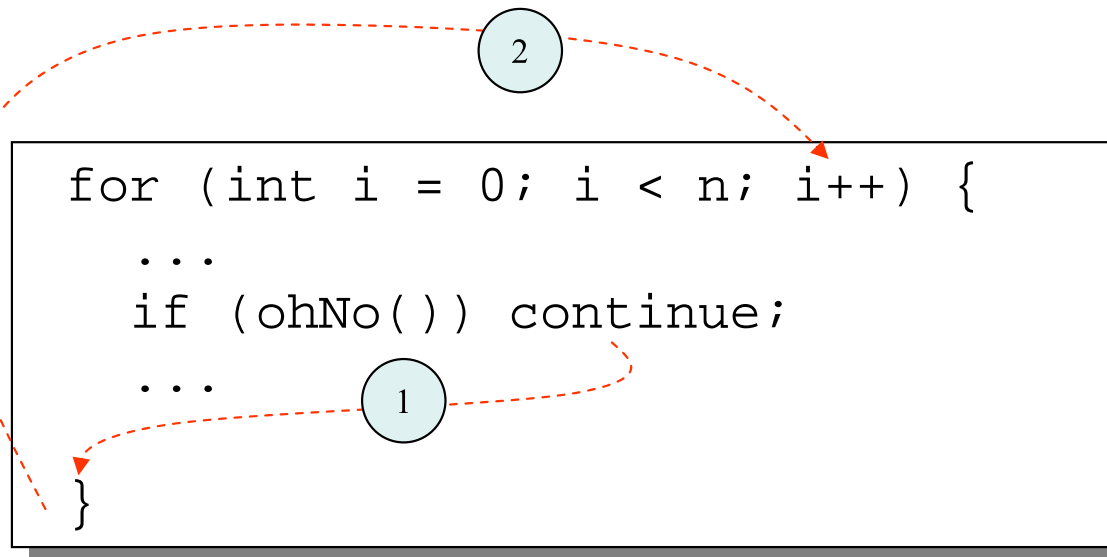
```
while (! success()) {  
    ...  
    if (ohNo()) continue;  
    ...  
}
```



```
do {  
    ...  
    if (ohNo()) continue;  
    ...  
} while (! success());
```



```
for (int i = 0; i < n; i++) {  
    ...  
    if (ohNo()) continue;  
    ...  
}
```



การประกาศและการสร้างอาเรย์

- ก่อนใช้อาเรย์ ต้อง

- ประกาศ ตัวแปรอาเรย์
- สร้าง ตัวอาเรย์

```
int [] data;
```

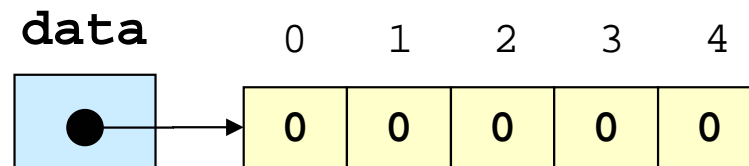
ประกาศตัวแปรชื่อว่า data มีไว้อ้างอิงอาเรย์ของ int

```
data = new int[5];
```

สร้างอาเรย์ของ int จำนวน 5 ช่อง แล้วให้ data เป็นตัวแปรสำหรับอ้างอิงอาเรย์ที่ได้สร้างขึ้น

```
int [] data = new int[5];
```

ประกาศตัวแปรและสร้างอาเรย์พร้อมกันเลยก็ได้



ข้อสังเกต : การประกาศ+การสร้าง

- การประกาศตัวแปรอาเรย์ ไม่ได้เป็นการสร้างตัวอาเรย์
- ขนาดของอาเรย์ถูกกำหนดตอนสร้างตัวอาเรย์ ไม่ใช่ตอนประกาศตัวแปร
- ขนาดของอาเรย์เป็นค่าของนิพจน์ก็ได้
(`data = new int[3*n + 1];`)
- อาเรย์ที่สร้างขึ้น ไม่สามารถเพิ่มหรือลดขนาดได้
- ระบบจะตั้งค่าเริ่มต้นของข้อมูลในอาเรย์โดยอัตโนมัติ
(ให้ค่าศูนย์กับอาเรย์ของจำนวน และ ให้ค่า `false` กับอาเรย์ของ `boolean`)

การใช้ข้อมูลในอาเรย์

- ใช้ `d[k]` แทนตัวที่ `k` ในอาเรย์ `d`
- index ของอาเรย์เริ่มที่ 0 เสมอ
- ถ้าอาเรย์มีขนาด `n` ตัว index ต้องเป็นจำนวนเต็ม มีค่าได้ตั้งแต่ 0 ถึง `n-1`

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

ประกาศ+สร้าง+ตั้งค่าเริ่มต้น

- ใช้ initializer list ระบุค่าเริ่มต้น
- ไม่ต้องกำหนดจำนวนช่องที่จะสร้าง
- จำนวนช่องที่สร้างเท่ากับจำนวนข้อมูลใน list

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

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

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

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

เขียนแยกเป็นสองบรรทัดแบบนี้ไม่ได้

ถ้าอยากรู้ขนาดของอาเรย์

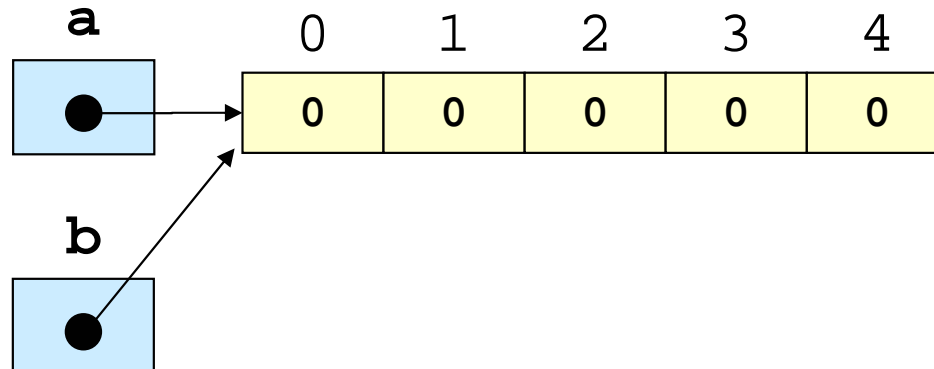
- เติม `.length` ตามหลังชื่ออาเรย์

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


ข้อสังเกต : การอ้างอิงอาร์เรย์

- อาร์เรย์แถวหนึ่ง อาจมีตัวแปรหลายตัวอ้างอิงได้

```
int [] a = new int[5];
```

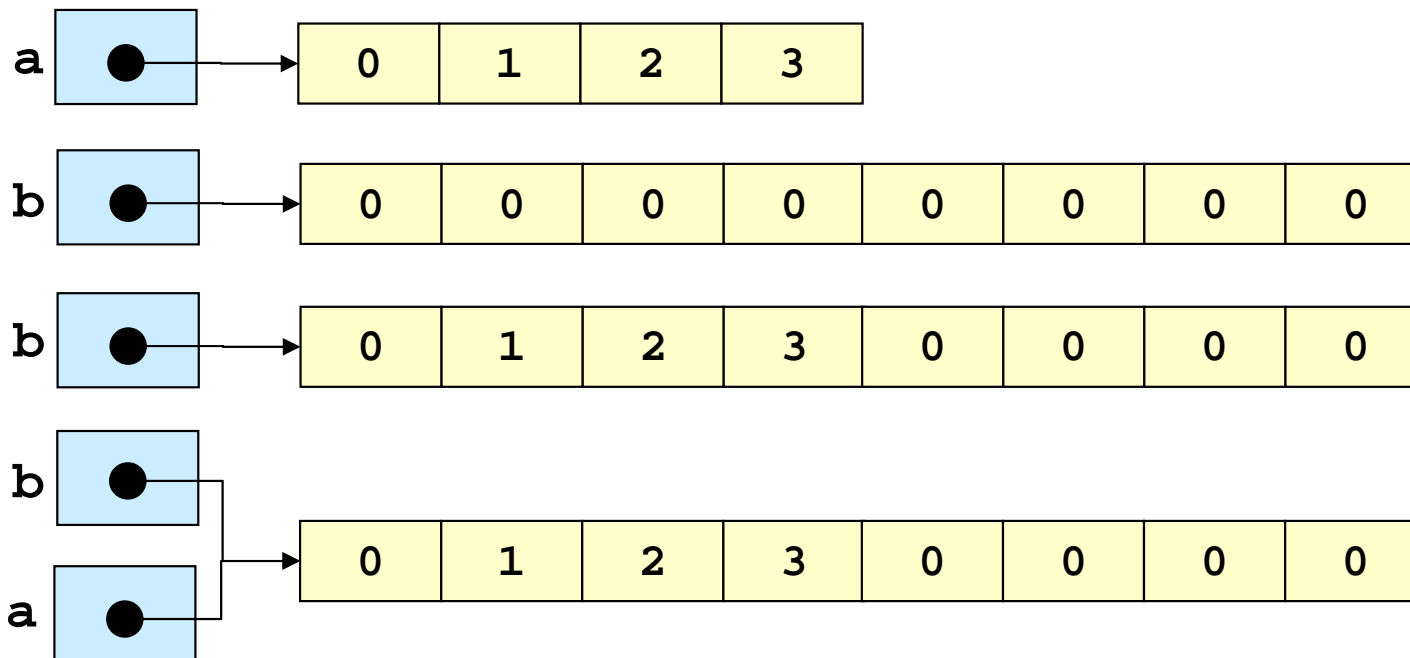


```
int [] b = a;
```

a กับ b มีค่าเท่ากัน หมายความว่าอ้างอิงอาร์เรย์เดียวกัน
ดังนั้น a[i] กับ b[i] อ้างอิงที่เก็บข้อมูลที่เดียวกัน

อยากเปลี่ยนขนาดอาเรย์ ก็ทำได้

```
int[] a = new int[4] {0,1,2,3};  
int[] b = new int[2 * a.length];  
for (int i = 0; i < a.length; i++) b[i] = a[i];  
a = b;
```



การอ้างอิงข้อมูลแบบผิดๆ

- ถ้าอาเรย์มีขนาด n ตัว
index ต้องเป็นจำนวนเต็มมีค่าได้ตั้งแต่ 0 ถึง $n-1$
- index ไม่ใช่จำนวนเต็ม - compile error
- index เป็นจำนวนเต็มที่อยู่นอกช่วง
 - เกิด **ArrayIndexOutOfBoundsException** ขณะทำงาน

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

```
C\>java Array2  
java.lang.ArrayIndexOutOfBoundsException  
    at Array2.main(Array2.java:4)  
Exception in thread "main"  
C\>
```

อาร์เรย์หลายมิติ (Multidimensional Array)

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

หนึ่งมิติ
(vector)

	0	1	2	3	4	5	6
a	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

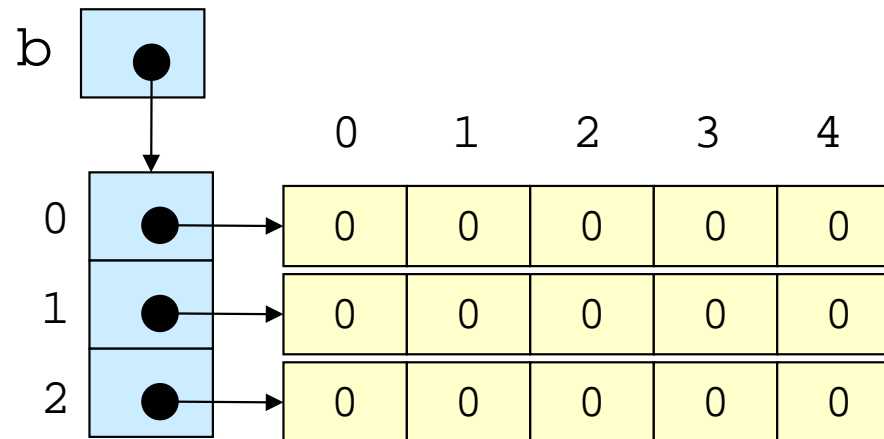
Initializer List

```
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

อาร์เรย์หลายมิติคืออาร์เรย์ของอาร์เรย์

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



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

Method : subprogram, function, procedure

- จาว่าเรียกว่า method

```
public class A {
    public static void main(String[] args) {
        int[] d = { 1, 2, 3, 4, 5, 6, 7 };
        int s = find( d, 5 );
        ...
    }
    static int find( int[] data, int x ) {
        int i;
        for (i = data.length-1; i >= 0; i--) {
            if (data[i] == x) break;
        }
        return i;
    }
}
```

Lab 1

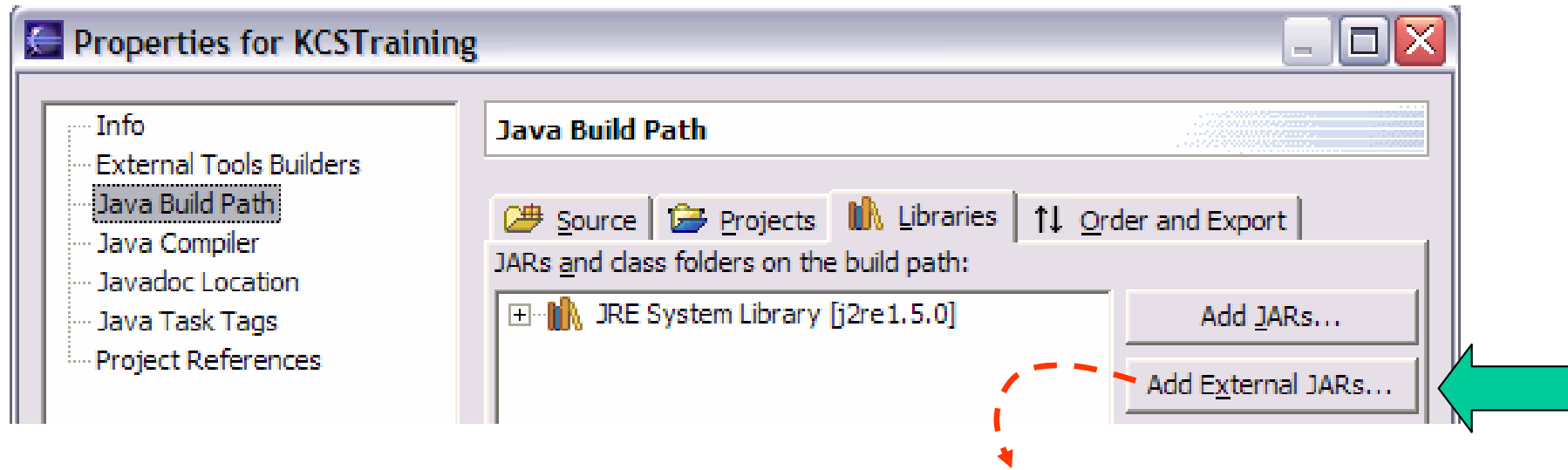
- มีข้อมูลกี่คู่ที่มีค่าแตกต่างกันอยู่ k ใน array 1 มิติ ?
- เช่น ให้ $k = 1$ และ array มีข้อมูลเป็น
10, 9, 12, 1, 8, 2
- จะได้ว่ามีข้อมูลอยู่ 3 คู่ที่มีค่าแตกต่างกันอยู่ 1 ดังนี้
(10,9), (9,8), (1,2) ดังนั้น diffByK จะคืนค่า 3

```
public class DiffByK {  
    ...  
    static int diffByK (int[] array, int k) {  
  
    }  
}
```

ทดสอบ : Run JUnit ของ lab1

ต้องเพิ่ม junit.jar ใน build path

Project -> Properties



<eclipse folder>\plugins\org.junit_3.8.1\junit.jar

