CHAPTER03

ชนิดข้อมูล ตัวแปร และตัวดำเนินการข้อมูล

บทนำ

- ในการเขียนโปรแกรมคอมพิวเตอร์สิ่งที่โปรแกรมต้องเข้าใจ คือ
- ตัวแปร
- ชนิดข้อมูล
- การประกาศ และการกำหนดค่าเริ่มต้นให้แก่ตัวแปร
- ใน chapter03 นี้เราจะได้ศึกษาสิ่งที่กล่าวมานี้

บทนำ

- ตัวดำเนินการทางคณิตศาสตร์และตัวดำเนินการเพื่อการกำหนดค่า
- ข้อมูลชนิด String และการจัดรูปแบบข้อมูลชนิดตัวเลข
- การใส่หมายเหตุในโปรแกรม
- การเขียนแบบเรียงลำดับ

ตัวแปร (variable)

- เป็นที่เก็บข้อมูลชั่วคราว
- ที่เก็บข้อมูลชั่วคราวในคอมพิวเตอร์ก็คือ RAM
- การที่เรียกตัวแปร ก็เพราะว่าสามารถเปลี่ยนแปลงค่าที่เก็บได้
- การเรียกใช้งานตัวแปรในภาษาจาวา จะต้องทำการประกาศตัวแปรก่อนเสมอ

หลักการตั้งชื่อให้แก่ Identifier

- จาวามีกฎในการตั้งชื่อให้กับ **Identifier** ต่างๆ ซึ่งได้แก่ ตัวแปร เมธอด คลาส แพ็กเกจ และอินเทอร์เฟส ดังนี้
- 1. ชื่อประกอบด้วยตัวอักษรภาษาอังกฤษ ตัวเลข **underscore (_)** หรือ **dollar sign ($)** ก็ได้ แต่จะต้องขึ้นต้นตัวอักษรภาษาอังกฤษ หรือ _ หรือ $ เท่านั้น
- 2. ชื่อที่ตั้งขึ้นมาจะมีช่องว่างไม่ได้

หลักการตั้งชื่อให้แก่ Identifier (ต่อ)

- 3. ชื่อที่ตั้งขึ้นเป็น **case-sensitive** หมายถึง ตัวอักษรตัวใหญ่กับตัวอักษรตัวเล็กถือว่าเป็นคนละตัวกัน เช่น **hello, HeLLo, HELLO** ทั้งหมดนี้ถือว่าเป็นคนละชื่อกัน
- 4. ชื่อที่ตั้งขึ้นต้องไม่ไปซ้ำกับคำสงวน **(reserved word)**

ตัวอย่างการตั้งชื่อ Identifier

Identifier ที่ถูกต้อง		Identifier ที่ผิด
firstname	bookTitle	12Street
lastname	sum	job Title
price	netPrice	phone@yahoo
_customer	empName	.lastName
stdid	salary	public
age	$name	C#

คำสงวน (reserved word)

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

white space in java

- white space หมายถึงตัวอักษรที่ไม่มีรูปร่างหน้าตา ซึ่งจาวาแบ่ง white space ออกเป็น 4 ชนิดได้แก่
- 1. horizontal tab คือ ตัวอักษรแท็บแนวนอน (\t)
- 2. form feed คือ ตัวอักษรขึ้นหน้าใหม่ (\f)
- 3. carriage return คือ ตัวอักษรเลื่อนเคอร์เซอร์ไปทางซ้ายสุด (\r)
- 4. linefeed หรือ newline คือ ตัวอักษรขึ้นบรรทัดใหม่ (\n)

white space in java (cont.)

- คอมไพเลอร์ของภาษาจาวาจะไม่สนใจ เราเขียนโปรแกรมทั้งหมดอยู่ภายในบรรทัดเดียวหรือเขียนแยกไว้หลายๆ บรรทัด
- เพราะ **white space** มีหน้าที่เพียงแต่ช่วยแยกคำให้อ่านง่ายขึ้นเท่านั้น
- ดังนั้นไม่ว่าจะเว้นช่องว่างระหว่างคำกี่ช่องก็ตาม ย่อมไม่มีผลต่อการตีความของคอมไพเลอร์
- หลักการของ **white space** คือ ต้องเว้นวรรคหลัง **reserved word**

Literal

- คือ ข้อมูลที่แน่นอน ซึ่งเราระบุตายตัวลงในโค้ดโปรแกรมว่าต้องให้มีค่าเป็นอะไร ตัวอย่างเช่น ‘A’, “Hello”, 100 เป็นต้น
- ภาษาจาวาได้แบ่ง **literal** ออกเป็น **5** ประเภท คือ
- 1. integer literal
- 2. floating-point literal
- 3. boolean literal
- 4. character literal
- 5. string literal

Integer Literal

- สามารถระบุ **integer literal** ได้ทั้งรูปแบบเลขฐานสิบ **(decimal)** เลขฐานสิบหก **(hexadecimal)** และเลขฐานแปด **(octal)**
- ถ้า **integer literal** เป็นเลขจำนวนเต็มแบบยาว **(long integer)** ต้องระบุ **L** หรือ **l** ต่อท้ายค่านั้น เช่น **35L, 149l** เป็นต้น
- ถ้า **integer literal** เป็นเลขฐานแปดให้นำหน้าค่านั้นด้วยเลข **0** เช่น **075, 0623** เป็นต้น
- ถ้า **integer literal** เป็นเลขฐานสิบหกให้นำหน้าค่านั้นด้วย **0x** (ศูนย์เอ็กซ์) เช่น **0x7B4, 0xF13** เป็นต้น

Floating-point Literal

- สามารถระบุ **floating-point literal** ได้ทั้งในรูปแบบเลขฐานสิบ เช่น 87.345 หรือในรูปแบบของ **exponential** เช่น 8.7345e1 (มีความหมายเท่ากับ 8.7345×10^1) เป็นต้น
- ถ้า **floating-point** เป็นเลขจำนวนจริงชนิด **float** ต้องระบุ F หรือ f ต่อท้ายค่านั้น เช่น 3.45F, 567.22f เป็นต้น
- ถ้า **floating-point** เป็นเลขจำนวนจริงชนิด **double** จะระบุ D หรือ d ต่อท้ายค่านั้นหรือไม่ก็ได้ เพราะค่า **default** ของ **floating-point literal** คือ **double**

Boolean Literal

- ประกอบไปด้วยค่าตรรกะ 2 ค่า เท่านั้น ก็คือ **true** (จริง) และ **false** (เท็จ)
- **boolean literal** นำไปใช้ส่วนการเขียนโปรแกรมที่มีการเปรียบเทียบ หรือโปรแกรมที่มีการตรวจสอบเงื่อนไข

Character Literal

- เป็นตัวอักษรตัวเดียว หรือจะเป็นค่า **escape sequence** ก็ได้ โดย **character literal** จะถูกคลุมด้วยเครื่องหมาย **single quote (')**
- ตัวอย่างเช่น 'A', 'a', '\\', '\u0027' เป็นต้น

Escape Sequence

Escape Sequence	ความหมาย
\b	ถอยหลัง 1 ช่องตัวอักษร (backspace)
\t	แท็บแนวนอน (hoirzontal tab)
\n	ขึ้นบรรทัดใหม่ (new line)
\f	ขึ้นหน้าใหม่ (form feed)
\r	เลื่อนเคอร์เซอร์ไปทางซ้ายสุด (carriage return)
\’	หมายถึงตัวอักษร ‘ (single quote)

Escape Sequence (cont.)

Escape Sequence	ความหมาย
\”	หมายถึงตัวอักษร “ (double quote)
\\	หมายถึงตัวอักษร \ (backslash)
\xxx	หมายถึงตัวอักษรที่มีในรหัสแอสกี้ (ASCII) เท่ากับค่าของเลขฐานแปดที่ระบุ (xxx คือ ตัวเลขในรูปฐานแปด) เช่น สมมติสั่งพิมพ์ค่า ‘\043’ ผลลัพธ์ที่ได้ คือ # เพราะตัวอักษร # มีรหัสแอสกี้เท่ากับ 43 ฐานแปด
\uxxxx	หมายถึงตัวอักษรที่มีค่าในรหัสยูนิโค้ด (Unicode) เท่ากับค่าของเลขฐานสิบหกที่ระบุ (xxxx คือตัวเลขในรูปเลขฐานสิบหก) เช่น สมมติสั่งพิมพ์ค่า ‘\u0023’ ผลลัพธ์ที่ได้คือ # เพราะตัวอักษร # มีรหัส Unicode เท่ากับ 23 ฐานสิบหก นั่นเอง

ตาราง escape sequence กับ unicode

Escape Sequence	Unicode
\b	\u000A
\t	\u0009
\n	\u0008
\f	\u000D
\r	\u000C
\’	\u0027
\”	\u0022
\\	\u005C

String Literal

- **String literal** อาจจะประกอบไปด้วยอักษรเพียงตัวเดียว ตัวอักษรหลายตัว หรือเป็นค่าสตริงว่าง (ไม่มีตัวอักษรใดๆ เลย) ก็ได้ โดย **string literal** จะถูกเขียนคลุมด้วยเครื่องหมาย **double quote (“”)**
- ภายในเครื่องหมาย **double quote** เราสามารถระบุ **escape sequence** เพื่อให้หมายถึงตัวอักษรหนึ่งๆ ได้
- ตัวอย่างเช่น **“”, “A”, “Java Programming”,**

“First Line\nSecond Line” เป็นต้น

ชนิดข้อมูลพื้นฐาน (Primitive Data Type)

- แบ่งได้ 4 กลุ่ม ดังนี้
- 1. Logical ได้แก่ boolean
- 2. Textual ได้แก่ char
- 3. Integer ได้แก่ byte, short, int และ long
- 4. Floating-point ได้แก่ float และ double

ชนิดข้อมูลพื้นฐาน (ที่เป็นตัวเลข)

ชนิดข้อมูล	จำนวนไบต์	ช่วงของข้อมูล
byte	1	-2^7 ถึง $2^7 - 1$ (-128 ถึง +127)
short	2	-2^{15} ถึง $2^{15} - 1$ ประมาณบวกลบสามหมื่น
int	4	-2^{31} ถึง $2^{31} - 1$ ประมาณบวกลบสองพันล้าน
long	8	-2^{63} ถึง $2^{63} - 1$ ประมาณบวกลบเก้าล้านล้านล้าน
float	4	$\pm 3.4 \times 10^{38}$ แม่นยำได้ประมาณ 6 ถึง 9 หลัก
double	8	$\pm 1.8 \times 10^{308}$ แม่นยำได้ประมาณ 15 ถึง 17 หลัก
char	16	

การประกาศและกำหนดค่าเริ่มต้นให้ตัวแปร

- รูปแบบการประกาศตัวแปร คือ

```
Data Type Variable;
```

- เมื่อ Data Type คือ ชนิดข้อมูล
- Variable คือ ชื่อตัวแปร

การประกาศและกำหนดค่าเริ่มต้นให้ตัวแปร

- รูปแบบการกำหนดค่าเริ่มต้นให้ตัวแปร คือ

```
Data Type Variable = Value Assigned;
```

- **Value Assigned** คือ ค่าที่ต้องการกำหนดเป็นค่าเริ่มต้นให้แก่ตัวแปร โดย **Value Assigned** ก็คือ **literal** ต่างๆ นั่นเอง

การประกาศและกำหนดค่าเริ่มต้นให้ตัวแปร

- เมื่อประกาศตัวแปรขึ้นแล้วเราต้องกำหนดค่าให้กับตัวแปรเอง จะไม่มีการกำหนดค่าเริ่มต้นให้กับตัวแปรโดยอัตโนมัติ
- หากมีการเรียกใช้งานตัวแปรที่ยังไม่ได้ถูกกำหนดค่า เมื่อคอมไพล์โปรแกรมแล้ว คอมไพเลอร์จะแจ้งข้อผิดพลาดว่า **“Variable might not have been initialized”**

Example 1

```
public class MyName {
    public static void main(String[] arg) {
        String myName = “Jonathan”;
        int myAge = 23;
        System.out.println(“My Name is “ + myName);
        System.out.println(“I am “ + myAge + “ year olds.”);
    }
}
```

การกำหนดค่าเริ่มต้นให้แก่ตัวแปร

ตัวดำเนินการ (Operator)

- ตัวดำเนินการ (operator) คือ สัญลักษณ์หรือเครื่องหมายแทนการกระทำกับข้อมูล เพื่อบอกให้เครื่องคอมพิวเตอร์ทราบว่าจะต้องดำเนินการใดกับข้อมูลใดบ้าง
- ภาษาจาวาแบ่งตัวดำเนินการได้เป็น
- 1. ตัวดำเนินการทางคณิตศาสตร์ Arithmetic Operator
- 2. ตัวดำเนินการทางการเปรียบเทียบ Comparison Operator
- 3. ตัวดำเนินการทางตรรกะ Logical Operator
- 4. ตัวดำเนินการทางบิต Bitwise Operator
- 5. ตัวดำเนินการเพิ่มค่า และลดค่า Increment & Decrement Operator
- 6. ตัวดำเนินการกำหนดค่า Assignment Operator

Arithmetic Operator

- ภาษาจาวามีเครื่องหมายทางคณิตศาสตร์ ดังตาราง

เครื่องหมาย	การดำเนินการ	ตัวอย่าง (x=10, y=7, z=2.5)	ผลลัพธ์
+	บวก	x+y;	17
-	ลบ	x-y;	3
*	คูณ	x*y;	70
/	หาร	x/z;	4.0
%	หารเอาเศษ (Modulus)	x%y;	3

Comparison Operator

- ตัวดำเนินการด้านการเปรียบเทียบในภาษาจาวา แสดงดังตาราง

เครื่องหมาย	การดำเนินการ	ตัวอย่าง (x=10, y=7, z=2.5)	ผลลัพธ์
>	มากกว่า	x>y	true
<	น้อยกว่า	x<y	false
>=	มากกว่าหรือเท่ากับ	z>=y	false
<=	น้อยกว่าหรือเท่ากับ	z<=x	true
!=	ไม่เท่ากัน	x!=y	true
==	เท่ากัน	x==10	true

Logical Operator

- ตัวดำเนินการทางตรรกะในภาษาจาวาแสดงดังตาราง

เครื่องหมาย	การดำเนินการ	ตัวอย่าง (a=true,b=false,c=true)	ผลลัพธ์
&&	AND	a&&c	true
||	OR	b||c	true
!	NOT	!a	false

Bitwise Operator

- ตัวดำเนินการด้าน **bit** ในภาษาจาวาแสดงดังตาราง

เครื่องหมาย	การดำเนินการ	ตัวอย่าง (x=10, y=7)	ผลลัพธ์
&	Bitwise AND	x&y	2
|	Bitwise OR	x|y	15
^	Bitwise Ex-OR	x^y	13
~	Bitwise NOT	~y	-8
>>	Shift Right	X>>2	2
<<	Shift Left	y<<2	28

Increment & Decrement Operator

- ตัวดำเนินการเพิ่มค่าและลดค่าแสดงดังตาราง

เครื่องหมาย	การดำเนินการ	ตัวอย่าง (x=10, y=7)	ผลลัพธ์
++	Increment	++x	11
		x++	12
--	decrement	--y	6
		y--	5

Assignment Operator

- ตัวดำเนินการกำหนดค่าก็คือเครื่องหมายเท่ากับ (=) ใช้สำหรับการกำหนดค่าให้แก่ตัวแปร
- รูปแบบการใช้งานตัวดำเนินการกำหนดค่า แบ่งได้ 3 กรณี ดังนี้

```
Variable = Constant
```

```
Variable = Variable
```

```
Variable = Expression
```

Assignment Operator (cont.)

- ตัวอย่างเช่น

```
a = 20;
```

```
b=a+30;
```

```
c=b;
```

Assignment Operator

- ตัวดำเนินการกำหนดค่าในภาษาจาวา แสดงดังตาราง

เครื่องหมาย	การดำเนินการ	ตัวอย่าง (x=10, y=7, z=0)	ผลลัพธ์
+=	กำหนดค่าบวก	z+=y (z=z+y)	7
-=	กำหนดค่าลบ	y-=x (y=y-x)	-3
*=	กำหนดค่าคูณ	y*=2 (y=y*2)	14
/=	กำหนดค่าหาร	x/=5 (x=x/5)	2
%=	กำหนดค่าหารเอาเศษ	x%=3 (x=x%3)	1

นิพจน์ (Expression)

- นิพจน์ (**expression**) คือ การนำเอาโอเปอเรเตอร์และโอเปอแรนด์หลายๆ ตัวมารวมกันเข้าเป็นประโยคเดียวกัน **และประมวลผลได้ผลลัพธ์ออกมา**
- โอเปอร์เรเตอร์ (**operator**) ก็คือตัวดำเนินการที่ได้กล่าวไปแล้ว
- โอเปอร์แรนด์ (**operand**) ก็คือตัวถูกดำเนินการ หรือตัวถูกกระทำ ซึ่งอาจจะเป็น **literal,** ตัวแปร หรือนิพจน์ก็ได้

นิพจน์ทางด้านคณิตศาสตร์

- นิพจน์ทางด้านคณิตศาสตร์ (Arithmetic Expression) ประกอบไปด้วยโอเปอร์แรนด์ที่ถูกกระทำด้วยตัวดำเนินการทางด้านคณิตศาสตร์ เช่น

```
X+Y*30/(45-33)
```

- การประมวลผลนิพจน์นี้จะต้องรู้ลำดับความสำคัญของเครื่องหมายทางคณิตศาสตร์เสียก่อนจึงจะทำการหาผลลัพธ์จากนิพจน์นี้ได้

นิพจน์ทางด้านคณิตศาสตร์

- ลำดับความสำคัญของเครื่องหมายทางด้านคณิตศาสตร์

ลำดับที่	เครื่องหมาย
1	()
2	++,--
3	*,/,%
4	+,-
5	=,+=,-=,*=,/=,%=

Data Type Conversion

- เมื่อต้องการดำเนินการทางคณิตศาสตร์ระหว่างโอเปอร์แรนด์ที่มีชนิดข้อมูลต่างกัน เราจะต้องทำการแปลงชนิดของข้อมูลของโอเปอร์แรนด์ให้เป็นชนิดเดียวกันก่อน จึงจะสามารถดำเนินการกันได้
- เราสามารถแบ่งประเภทการแปลงชนิดข้อมูลได้ 2 ประเภท คือ
- **1. Implicit type conversion** เป็นการแปลงชนิดข้อมูลที่ภาษาจาวาทำให้โดยอัตโนมัติ
- **2. Explicit type conversion** (หรือ **Casting**) เป็นการแปลงชนิดข้อมูลที่ผู้เขียนโปรแกรมต้องกระทำเอง

Data Type Conversion (cont.)

- รูปแบบการแปลงชนิดข้อมูลด้วย **explicit type conversion** คือ

(Data Type) นิพจน์ที่ต้องการแปลงชนิดข้อมูล

- การทำ **explicit type conversion** นั้น ปกติเราจะแปลงจากชนิดข้อมูลที่มีนัยสำคัญต่ำกว่าไปเป็นชนิดข้อมูลที่มีนัยสำคัญสูงกว่า

Data Type Conversion (cont.)

- ตารางแสดงชนิดข้อมูลที่มีนัยสำคัญจากสูงสุดไปต่ำสุด

data type	significant
double	สูงสุด
float	
long	
int	
short	
byte	ต่ำสุด

Example 2

```
public class TestSignificant {
    public static void main(String[] arg) {
        float a=1.0f,b=2.5f;
        int c;
        double d;
        d = a+b;
        System.out.println("d = " + d);
        c = (int) (a+b);
        System.out.println("c = " + c);
    }
}
```

implicit conversion

explicit conversion

Test

- ตอบคำถามต่อไปนี้
- 1) 5/2 เท่ากับเท่าใด? 2
- 2) 5.0/2.0 เท่ากับเท่าใด? 2.5
- 3) 5.0/2 เท่ากับเท่าใด? 2.5
- 4) 5/2.0 เท่ากับเท่าใด? 2.5

Summary

- การกระทำนิพจน์ทางด้านคณิตศาสตร์ระหว่างจำนวนเต็มกับจำนวนทศนิยม มีบทสรุปดังนี้

จำนวนเต็มกระทำกับจำนวนเต็มจะได้ **จำนวนเต็ม**
จำนวนจริงกระทำกับจำนวนจริงจะได้ **จำนวนจริง**
จำนวนจริงกระทำกับจำนวนเต็มจะได้ **จำนวนจริง**
จำนวนเต็มกระทำกับจำนวนจริงจะได้ **จำนวนจริง**

Comparison Expression

- นิพจน์ด้านการเปรียบเทียบ ประกอบไปด้วยโอเปอร์แรนด์ที่ถูกกระทำโดยตัวดำเนินการทางด้านการเปรียบเทียบ เช่น

```
30 > 20
```

```
40 == 20
```

- ผลลัพธ์ของการเปรียบเทียบมีค่าที่เป็นไปเพียงได้ 2 อย่างคือ **true** และ **false** เท่านั้น

Logical Expression

- นิพจน์การดำเนินการทางด้านตรรกะ **(logical expression)** ประกอบไปด้วยโอเปอร์แรนด์ที่ถูกกระทำโดยตัวดำเนินการทางด้านตรรกะ
- โดยทั่วไปโอเปอร์แรนด์ของตัวดำเนินการทางด้านตรรกะจะมีค่าเป็น **boolean literal**
- ผลลัพธ์ที่ได้จากนิพจน์ทางด้านตรรกะก็มีได้ 2 ค่าเท่านั้นคือ **true** และ **false**

AND Operator

- ตารางการทำงานของตัวดำเนินการ AND Operator

A	B	Y=A&&B
false	false	false
false	true	false
true	false	false
true	true	true

OR Operator

- ตารางการทำงานของตัวดำเนินการ OR Operator

A	B	Y=A||B
false	false	false
false	true	true
true	false	true
true	true	true

NOT Operator

- ตารางการทำงานของตัวดำเนินการ NOT Operator

A	Y=!A
false	true
true	false

ค่าคงที่ (constant)

- ค่าคงที่ คือตัวแปรที่เมื่อกำหนดค่าแล้ว จะมีค่าเป็นค่านั้นไปตลอดการทำงานของโปรแกรม ไม่สามารถเปลี่ยนแปลงค่าได้
- การประกาศค่าคงที่มีลักษณะคล้ายกับการประกาศตัวแปร เพียงแต่จะต้องระบุคีย์เวิร์ด **final** ด้วย มีรูปแบบ ดังนี้

```
final data type variable = value assigned;
```

- ตัวอย่าง

```
final double PI = 3.1416;
```

ตัวดำเนินการบวก (+) กับ String Literal

- สตริง (String) รู้กันทั่วๆ ไปว่าคือข้อความ ที่ประกอบไปด้วยด้วยอักขระต่างๆ (a-z, A-Z, special character, etc.)
- โดยสตริงจะเขียนอยู่ภายใต้เครื่องหมาย double quote (“”)
- ตัวดำเนินการบวก (+) ที่นำมาใช้กับข้อมูลที่เป็นสตริง จะเรียกว่า ตัวเชื่อมสตริง (String Concatenation)

ตัวดำเนินการบวก (+) กับ String Literal

- พิจารณาคำสั่งต่อไปนี้ ว่ามีผลลัพธ์คืออะไร?
- System.out.println(“4+2=“);
- System.out.println(4+2);
- System.out.println(“4+2=“ +4+2);
- System.out.println(“4+2=“ + (4+2));

4+2=
6
4+2=42
4+2=6

การเขียนโปรแกรมแบบเรียงลำดับ

- เป็นการเขียนโปรแกรมด้วยประโยคคำสั่งต่างๆ ที่เรียงลำดับต่อเนื่องกันไป จากคำสั่งหนึ่งไปยังอีกคำสั่งหนึ่ง
- โปรแกรมจะเริ่มทำงานที่ประโยคคำสั่งแรกให้เสร็จสิ้นก่อนจึงสามารถทำงานตามคำสั่งต่อไปได้
- โปรแกรมที่มีการเขียนประโยคคำสั่งในลักษณะนี้จะเป็นโปรแกรมที่ไม่มีความซับซ้อนและไม่มีการตัดสินใจ

Example 3

- ให้นักศึกษาลองเขียนโปรแกรมแบบเรียงลำดับจากโจทย์ต่อไปนี้
- จงเขียนโปรแกรมแปลงค่าเงินบาทเป็นเงินดอลลาร์สหรัฐ (ปัจจุบันค่าเงินบาทต่อ 1 ดอลลาร์สหรัฐ เท่ากับ 32.538 บาท) โดยระบุจำนวนเงินบาทที่ต้องการจะแปลงเป็นเงินดอลลาร์สหรัฐ

เอกสารอ้างอิง

อรพิน ประวัติบริสุทธิ์. คู่มือการเขียนโปรแกรมด้วยภาษา Java. กรุงเทพฯ : โปรวิชั่น จำกัด, 2537.

พนิดา พาณิชกุล. การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษา Java. กรุงเทพฯ : เค ที พี คอมพ์ แอนด์ คอนซัลท์, 2548.

End.