Part 3

Java Programming Language
Mr.Rungrote Phonkam
rungrote@it.kmitl.ac.th

Contents

1. Identify

กฎการตั้งชื่อ

- ใช้ตั้งชื่อ Class, Data, Method, Variable, Label, …
- ประกอบด้วยตัวอักษร และหรือตัวเลข โดยตัวอักษรให้ใช้ตัวอักษรภาษาอังกฤษไม่ว่าตัวเลขหรือตัวใหญ่ รวมถึงสัญลักษณ์พิเศษ _ หรือ $ เช่น age, name2, int2float, _name, Currency$ เป็นต้น
- ความยาวตัวอักษรไม่ควรเกิน 65535 ตัวอักษร
- ไม่ควรมีตัวเลขเป็นตัวแรก เช่น 101database, 2name ถือว่าไม่สามารถใช้ตั้งชื่อได้
- ตัวอักษรตัวเล็กและตัวใหญ่มีความแตกต่างกัน ดังนั้น Count, count และ CoUnT ทั้งสามตัวอ่านเหมือนกัน แต่ถือว่าเป็นคนละตัวกัน

1. Identify

กฎการตั้งชื่อ(ต่อ...)

- ต้องไม่ตรงกับคีย์เวิร์ดใดในภาษาจาวาต่อไปนี้

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

1. Identify

การตั้งชื่อตามรูปแบบ

เมื่อตั้งชื่อคลาส ในแต่ละคำกำหนดให้พยัญชนะตัวแรกเป็นตัวอักษรตัวพิมพ์ใหญ่ ตัวอักษรที่เหลือในคำเดียวกันให้เป็นตัวอักษรตัวพิมพ์เล็ก เช่น MyClass, HelloJava, CollectedData เป็นต้น

เมื่อตั้งชื่อดาต้าหรือตัวแปร ตัวอักษรทุกตัวเป็นตัวอักษรตัวพิมพ์เล็กทั้งหมด เช่น color, name, height เป็นต้น

1. Identify

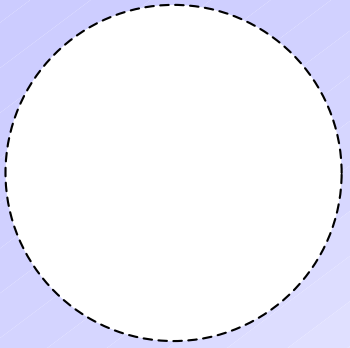
กฎการตั้งชื่อ (ต่อ...)

เมื่อตั้งชื่อดาต้าตัวแปรสำหรับเก็บข้อมูลคงที่ ชื่อนี้มักจะมีคีย์เวิร์ด final กำกับที่ด้านหน้า ให้ใช้ตัวอักษรทุกตัวเป็นตัวอักษรตัวพิมพ์ใหญ่ทั้งหมด เช่น SIZE, AMOUNT เป็นต้น

เมื่อตั้งชื่อเมธรอด กำหนดให้แรกในชื่อเมธรอดเป็นตัวอักษรตัวพิมพ์เล็ก ส่วนในคำถัดไปจากนั้นให้พยัญชนะตัวแรกให้เป็นตัวอักษรตัวพิมพ์ใหญ่ ตัวอักษรที่เหลือในคำเดียวกันให้เป็นตัวอักษรตัวพิมพ์เล็ก เช่น countMoney, openFile, setSize, print เป็นต้น

1. Identify

```
class SimpleClass  {
}
```




```
java SimpleClass
Exception in thread "main" java.lang.NoSuchMethodError: main
```

2. Data Member

รูปแบบ

Access_Level ***final*** ***static*** Data_Type Data_Name

Access_Level	คือระดับในการเข้าถึงเพื่อใช้งาน ไม่ระบุหมายถึง package
final	คือการกำหนดให้เป็นค่าคงที่
static	คือการกำหนดให้มีลักษณะ static
Data_Type	คือชนิดของข้อมูล
Data_Name	คือชื่อของดาต้า (ตั้งตามกฎการตั้งชื่อ)

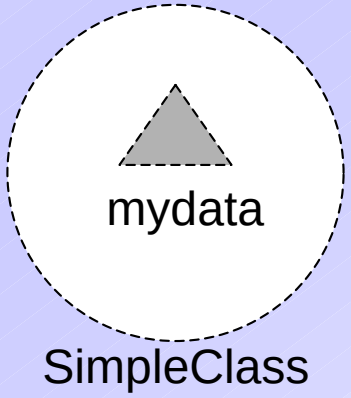
2. Data Member

ระดับการเข้าถึงเพื่อใช้งาน (Access Level)

Keyword	Class	Instance	Extended_Class	Extended_Instance	Extended_Class Outer Package	Extended_Instance Outer Package
public	✓	✓	✓	✓	✓	✓
protected	✓	✓	✓	✓	✓	
package*	✓	✓	✓	✓		
private	✓					

2. Data Member

```
class SimpleClass {
    int mydata;
}
```




```
java SimpleClass
Exception in thread "main" java.lang.NoSuchMethodError: main
```

3. Method Member

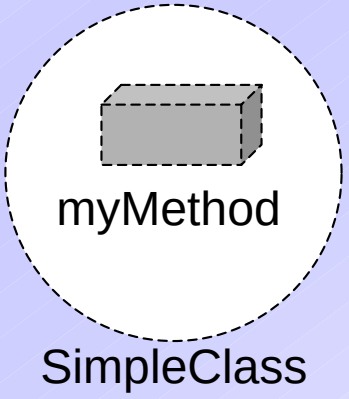
รูปแบบ

```
Access_Level  final  static Return_Type  Method_Name ( Argument_List )
{
        Statements
}
```

Access_Level	คือระดับในการเข้าถึงเพื่อใช้งาน
final	คือการกำหนดให้เป็นค่าคงที่
static	คือการกำหนดให้มีลักษณะ static
Return_Type	คือชนิดของข้อมูลที่ใช้คืนค่าจากเมธรอด
Method_Name	คือชื่อของเมธรอด (ตั้งตามกฎการตั้งชื่อ)
Argument_List	คืออาร์กิวเม็นต์ที่ส่งให้เมธรอดทำงาน
Statements	คือคำสั่ง(สเตจเม็นต์)ที่ใช้ทำงานในเมธรอด

3. Method Member

```
class SimpleClass {
    void myMethod( )
    {
    }
}
```




```
java SimpleClass
Exception in thread "main" java.lang.NoSuchMethodError: main
```

4. Literals

ข้อมูลค่าคงที่

- ข้อมูลที่ถูกเขียนลงในซอร์สโค้ดโดยตรง
- ชนิดตัวเลขจำนวนเต็ม

  เช่น 1, 46 หรือ 7048

  – เมื่อนำหน้าด้วยเลข 0 และตามหลังด้วยเลข 0 ถึง 7 หมายถึงเลขฐานแปด

  เช่น 024 มีค่าเท่ากับ 24_8 หรือ 30 ในฐานสิบ

  – เมื่อนำหน้าด้วยเลข 0x หรือ 0X และตามหลังด้วยเลข 0 ถึง 9 หรืออักษร a ถึง z หรืออักษร A ถึง Z หมายถึงเลขฐานสิบหก

  เช่น 0x1D หรือ 0X1d มีค่าเท่ากับ $1D_{16}$ หรือ 29 ในฐานสิบ

4. Literals

- ชนิดตัวเลขทศนิยม

  เช่น 12.4, 8.0, 9.33333 หรือ 24E5

  – เมื่อต้องการระบุพื้นที่เป็นชนิด float ให้ใช้ตัวอักษร F หลังตัวเลข
  เช่น 3.0F ทำให้มีพื้นที่ในการเก็บเลข 32 บิต

  – เมื่อต้องการระบุพื้นที่เป็นชนิด double ให้ใช้ตัวอักษร D หลังตัวเลข
  เช่น 3.0D ทำให้มีพื้นที่ในการเก็บเลข 64 บิต

  – ดังนั้น 3.0F และ 3.0D มีค่าเท่ากันก็ตาม แต่ต่างกันที่พื้นที่ในการบันทึก
  3.0F ใช้พื้นที่ขนาด 32 บิต
  3.0D ใช้พื้นที่ขนาด 64 บิต

  – ในกรณีที่ไม่ระบุ F หรือ D จะถือว่าเป็นแบบ double

4. Literals

- ชนิดตัวอักษร

  การกำหนดต้องอยู่ในสัญลักษณ์ (Single Quote)

  เช่น '4' คือ**ตัวอักษรเลขสี่**

  'A' คือ**ตัวอักษรเอใหญ่**

  '9' คือ**ตัวอักษรเลขเก้า**

  '+' คือ**ตัวอักษรสัญลักษณ์บวก**

4. Literals

- ชนิดข้อความ

  การกำหนดต้องอยู่ในสัญลักษณ์ (Double Quote)

  เช่น "Java", ”Hello! World", ”A", ”9",

  ข้อสังเกตสำหรับข้อมูลค่าที่ชนิดข้อความ

  – "" หมายถึงข้อมูลค่าคงที่ชนิดข้อความที่ไม่บรรจุตัวอักษรใดๆ หรือที่เรียกว่าคือ NULL

  – " " หมายถึงข้อมูลค่าคงที่ชนิดข้อความที่บรรจุตัวอักษรช่องว่าง (Space Character)

  – "A" หมายถึงข้อมูลค่าคงที่ชนิดข้อความที่บรรจุตัวอักษร A ซึ่งไม่เท่ากับ 'A'

4. Literals

- ชนิดตรรกะ ใช้สำหรับกำหนดค่าทางตรรกะ

true	คือข้อมูลค่าความเป็นจริง
false	คือข้อมูลค่าความเท็จ

4. Literals

การใช้งานข้อมูลค่าคงที่

- กำหนดค่าเริ่มต้นให้ดาต้าหรือตัวแปร

```
int i = 10 ;
float f = 51.4 ;
char ch = 'j' ;
```

- ใช้สร้างเอ็กเพลสชั่น

```
b  == true ;
status < 75 ;
k = k + 4 ;
```

5 Data Type

ชนิดข้อมูล

- ชนิดพลีมิทีฟ (Primitive Data Type)
  - เหมือนที่ใช้ใน C Language
  - int, char, float, double, ...
- ชนิดอ้างถึง (Reference Data Type)
  - Instance / Object
  - Array

5.1. Primitive Data Type

ชนิดข้อมูลพลิมีทีฟ

- ชนิดตัวเลขจำนวนเต็ม ใช้สำหรับเก็บข้อมูลที่เป็นเลขจำนวนเต็ม โดยใช้คีย์เวิร์ดคือ byte, short, int และ long
- ชนิดตัวเลขทศนิยม ใช้สำหรับเก็บข้อมูลที่เป็นเลขจำนวนทศนิยม ดังนั้นข้อมูลชนิดนี้มีความละเอียดกว่าตัวเลขจำนวนเต็ม โดยใช้คีย์เวิร์ด float และ double
- ชนิดตัวอักษร ใช้สำหรับเก็บข้อมูลที่ตัวอักษร หรือพยัญชนะหนึ่งตัว โดยใช้คีย์เวิร์ด char
- ชนิดตรรกะ ใช้สำหรับเก็บข้อมูลที่เป็นความจริง หรือความเท็จ โดยใช้คีย์เวิร์ด boolean
- ชนิดไม่มีผลใด ใช้สำหรับการกำหนดว่าไม่มีข้อมูลใดๆ หรือไม่มีการบันทึกสิ่งใดๆ void

5.1. Primitive Data Type

คีย์เวิร์ดสำหรับชนิดข้อมูล

Keyword	ข้อมูล	ขนาด(บิต)	ค่าเริ่มต้น
byte	จำนวนเต็ม	8	0
short		16	0
int		32	0
long		64	0
float	ทศนิยม	32	0.0
double		64	0.0
char	ตัวอักษร	16	‘’
boolean	ตรรกะ	true/false	false
void		-	-

5.2. Reference Data Type

ชนิดข้อมูลอ้างถึง

- เมื่อมีการกำหนดโดยใช้ชื่อคลาสใดๆ รวมถึงคลาสที่สืบทอดมาด้วย (ซึ่งสังเกตุได้จากชื่อคลาส ตัวอักษรตัวแรกของคำเป็นตัวใหญ่) เช่น Integer, StreamInput, Applet, Printer เป็นต้น
- ชนิดข้อมูลอ้างถึงสร้างขึ้นได้เอง โดยการสร้างคลาสขึ้นมาก่อน
- เมื่อมีการกำหนดข้อมูลเป็นลักษณะของอะเรย์ (Array)

(ชนิดข้อมูลอ้างถึงจะยกไปกล่าวอีกทีในภายหลัง)

6 Variables

ตัวแปร

- Class Variable หรือ Data Member (กล่าวผ่านมาแล้ว ก่อนหน้านี้) คือตัวแปรที่ถูกใช้ในคลาส สำหรับเก็บข้อมูลภายในแต่ละคลาส ในบทเรียนนี้เรียกตัวแปรในคลาสว่า **Data** หรือ **Data Member** เพื่อไม่ให้สับสนกับตัวแปรแบบอื่นๆ
- Method Variable คือตัวแปรที่ถูกกำหนดขึ้นภายในเมธรอด และถูกใช้งานได้ภายในเมธรอดเท่านั้น ในบทเรียนนี้เรียกว่า **ตัวแปร** หรือ **Variable**
- Parameter Variable คือตัวแปรที่ถูกใช้งานในเมธรอด แต่เกิดจากการผ่านข้อมูลเข้ามาในช่อง พารามิเตอร์ ในบทเรียนนี้จะเรียกว่า **Parameter**

6.1. Class Variables

Data Member

- กำหนดได้ทั้งจากชนิดข้อมูล Primitive หรือ Reference

```
class Class_Name {
        Data_Member
        Data_Member
        ...
        Data_Member
}
```

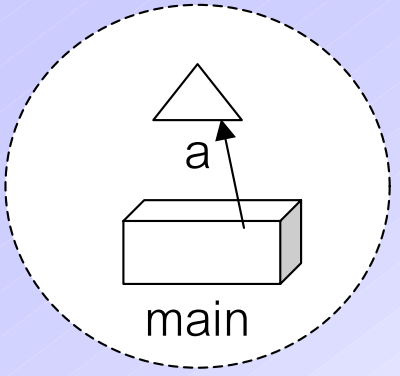
- การสร้างคาต้าด้วยชนิดข้อมูลพลีมิทีฟ

รูปแบบ

Access_Level ***final*** ***static*** Primitive_Data_Type Data_Name

6.1. Class Variables

```
class PublicStaticData  {
        public static int a = 10;
        public static void main(String args[])  {
                System.out.println(a);
        }
}
```




```
java PublicStaticData
10
```

6.1. Class Variables

เมธรอดเกี่ยวกับการแสดงผลลัพธ์

System.out.print
สำหรับพิมพ์ข้อมูลที่จอภาพ (Output)

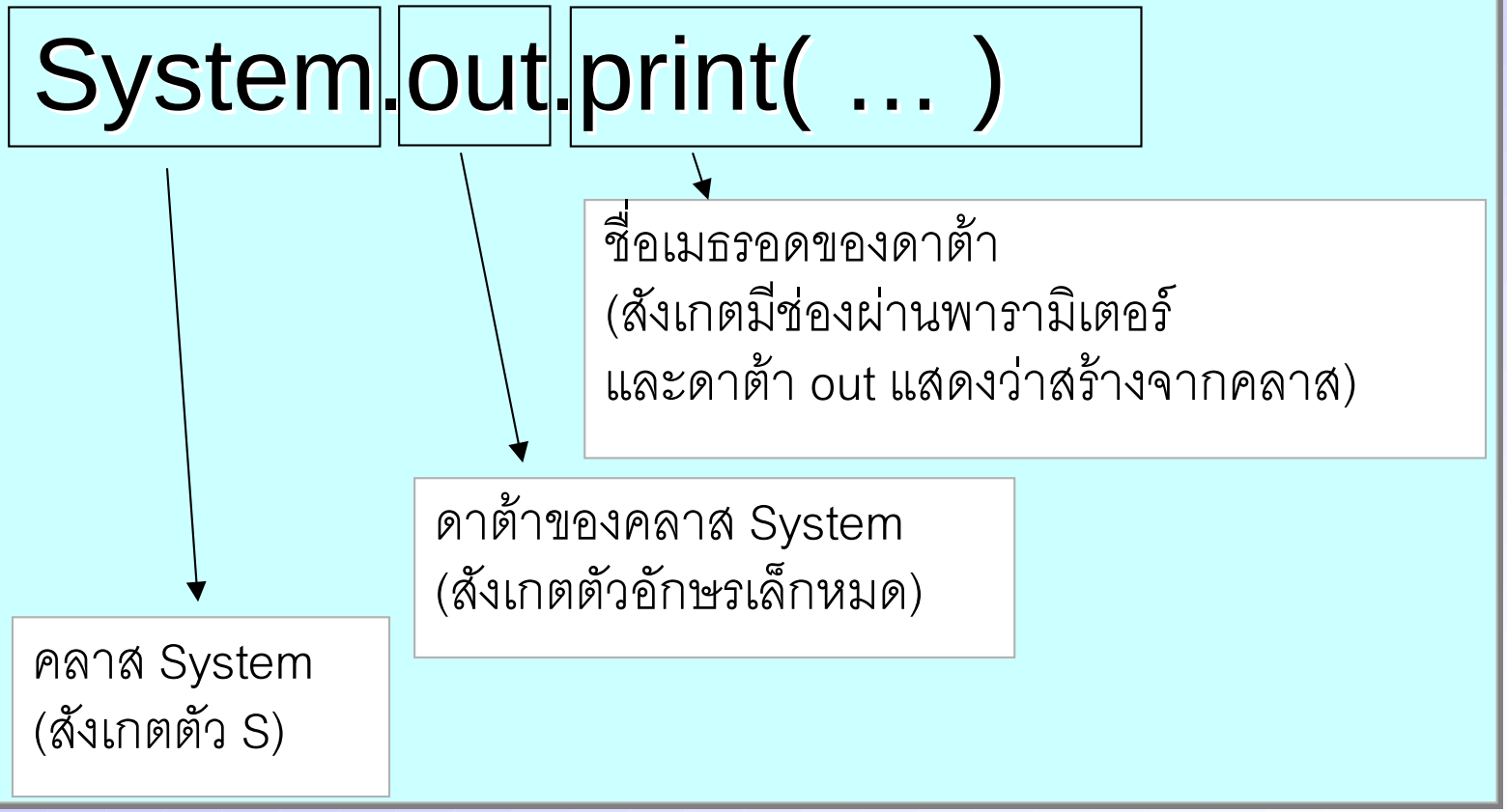
System.out.println
สำหรับพิมพ์ข้อมูลที่จอภาพ (Output) โดยมีการเลื่อนเคอร์เซอร์ไปที่บรรทัดใหม่

System.err.print
สำหรับพิมพ์ข้อมูลที่ส่วนแสดงข้อผิดพลาด (Error) ซึ่งโดยส่วนใหญ่จะใช้จอภาพแสดงข้อผิดพลาดเช่นเดียวกับใช้แสดงข้อมูล

System.err.println
สำหรับพิมพ์ข้อมูลที่ส่วนแสดงข้อผิดพลาด (Error) โดยมีการเลื่อนเคอร์เซอร์ไปที่บรรทัดใหม่

6.1. Class Variables

ตีความหมายเมธรอด System.out.print



6.1. Class Variables

ตัวอย่างการใช้งาน

System.out.println(“Hello Java”)

หมายถึงการพิมพ์ข้อความ Hello Java ที่จอภาพ

System.out.println(“Hello” + “ Java”)

หมายถึงการพิมพ์ข้อความ Hello Java ที่จอภาพ โดยสัญลักษณ์ + ใช้สำหรับการรวมข้อความ Hello กับข้อความ Java

System.out.println(a)

หมายถึงการพิมพ์ข้อความจากข้อมูลในตัวแปร a ถ้า a สร้างมาจากตัวเลขจำนวนเต็มข้อมูลก็แสดงเป็นเลขจำนวนเต็ม

6.1. Class Variables

ตัวอักษรควบคุม

\? เมื่อ ? หมายถึงตัวอักษรที่ใช้แทนรหัสควบคุม ถูกใช้ในชนิดข้อมูลตัวอักษรหรือข้อความ

เช่น

\n	หมายถึง Linefeed
\t	หมายถึง Tab
\r	หมายถึง Carriage Retuen
\f	หมายถึง Formfeed
\b	หมายถึง Backspace
\\	หมายถึง Backslash
\'	หมายถึง Single Quote
\"	หมายถึง Double Quote

6.1. Class Variables

- **Octal Escape Code**

  \nnn เมื่อ nnn หมายถึงตัวเลขฐานแปด

  เช่น

  \0 \277 \377

- **Unicode Escape Code**

  \uxxxx เมื่อ xxxx หมายถึงตัวเลขฐานสิบหก

  เช่น

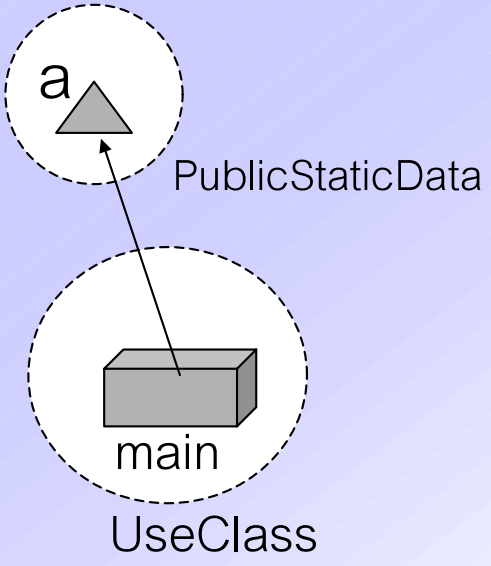
  \u0041 คือตัวอักษร A เป็นต้น

6.1. Class Variables

```
class TestFinalData  {
    static final char c = 'J';
    public static void main(String args[])  {
            System.out.println(a);
            c = 'a';
    }
}
```

6.1. Class Variables

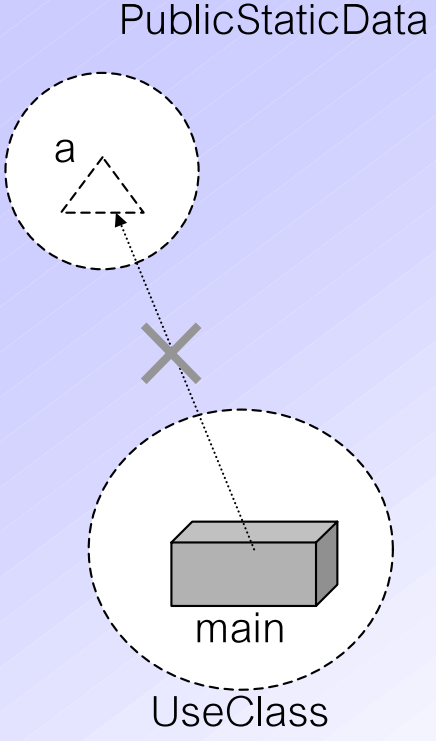
```
class PublicStaticData  {
    public static int a = 10;
}
class UseClass {
    public static void main(String args[])
    {   System.out.println(PublicStaticData.a);
    }
}
```




```
Java UseClass
10
```

6.1. Class Variables

```
class PublicStaticData  {
    public int a = 10;
}
class UseClass  {
    public static void main(String args[])
    {       System.out.println(PublicStaticData.a);
    }
}
```



6.1. Class Variables

Data Member

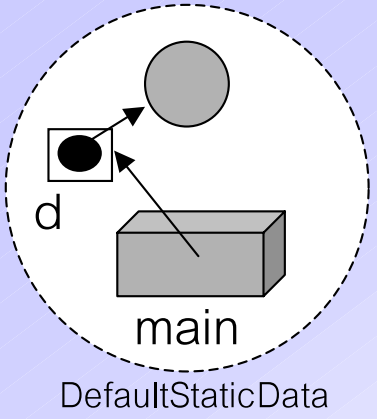
- การสร้างคาต้าด้วยชนิดข้อมูลอ้างถึง (จากคลาส)

รูปแบบ *Access_Level* ***final*** ***static*** Class_Name Data_Name
Data_Name = new Constructor_Name

ตัวอย่าง *public* ***final*** ***static*** Double myDouble;
myDouble = new MyDouble(10.0);
หรือ *public* ***final*** ***static*** Double myDouble = new MyDouble(10.0);

6.1. Class Variables

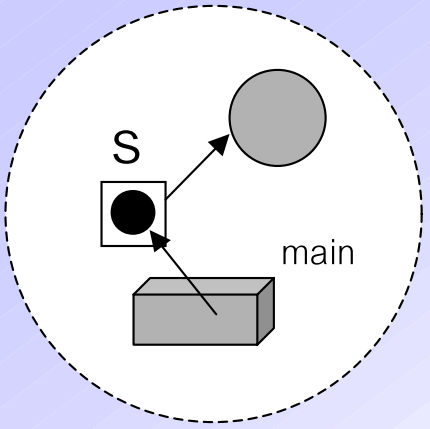
```
class DefaultStaticRefData {
    static Double d = new Double(255.53);
    public static void main(String args[])
    {       System.out.println(d.toString());
    }
}
```




```
Java DefaultStaticRefData
255.53
```

6.1. Class Variables

```
class PublicStaticRefData {
        public static String s = “Hello”;
        public static void main(String args[])
        {       System.out.println(s);
        }
}
```




```
Java PublicStaticRefData
Hello
```

6.2. Method Variables

ตัวแปรภายในเมธรอด

- เป็นได้ทั้งแบบ Primitive และแบบ Reference

```
class Class_Name {
        Method_Name()  {
                Method Variable
                Method Variable
                ...
        }
}
```

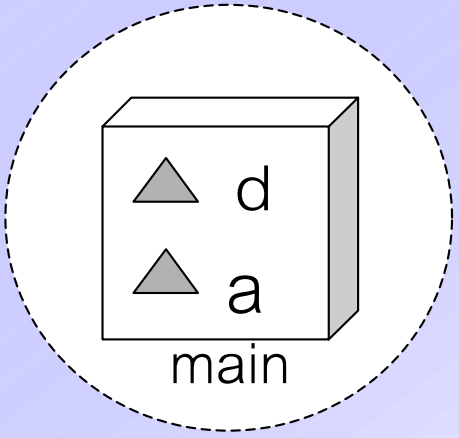
- การสร้างตัวแปรด้วยชนิดข้อมูลพลีมิทีฟ

รูปแบบ

final Primitive_Data_Type Data_Name

6.2. Method Variables

```
class PrimitiveVariable  {
    public static void main(String args[])  {
                double d;
                d = 11.11;
                System.out.println(d);
                int a = 3500;
                System.out.println(a);
        }
}
```



PublicStaticData

```
java PrimitiveVariable
11.11
3500
```

6.2. Method Variables

ตัวแปรภายในเมธรอด

- แบบ Reference

รูปแบบ

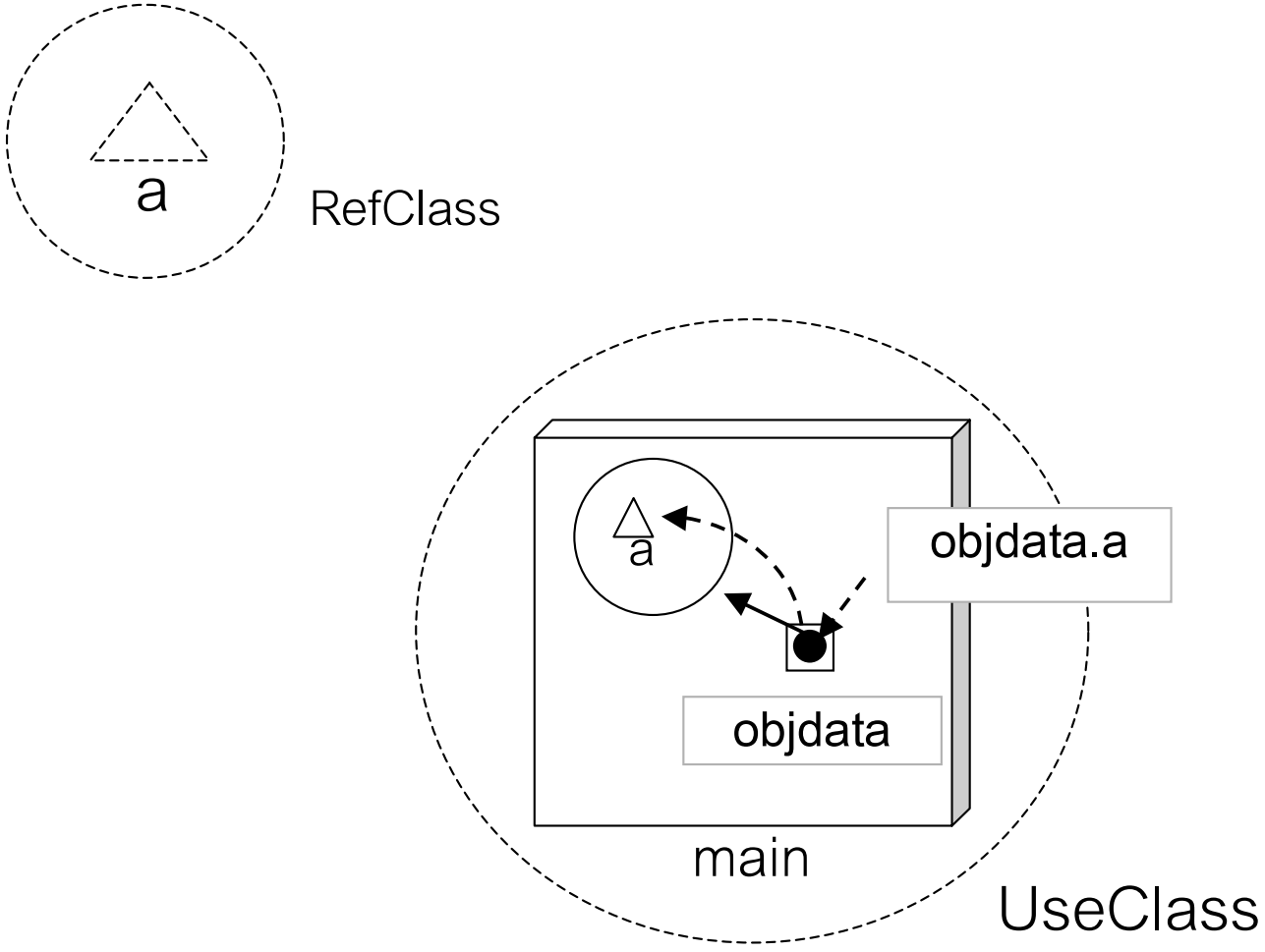
```
final  Class_Name  Data_Name = new Constructor()
```

6.2. Method Variables

```
class RefClass  {   public int a = 10;        }
class UseClass  {
    public static void main(String args[])  {
            RefClass objdata = new RefClass ();
            System.out.println(objdata.a);
    }
}
```

```
java UseClass
10
```

6.2. Method Variables

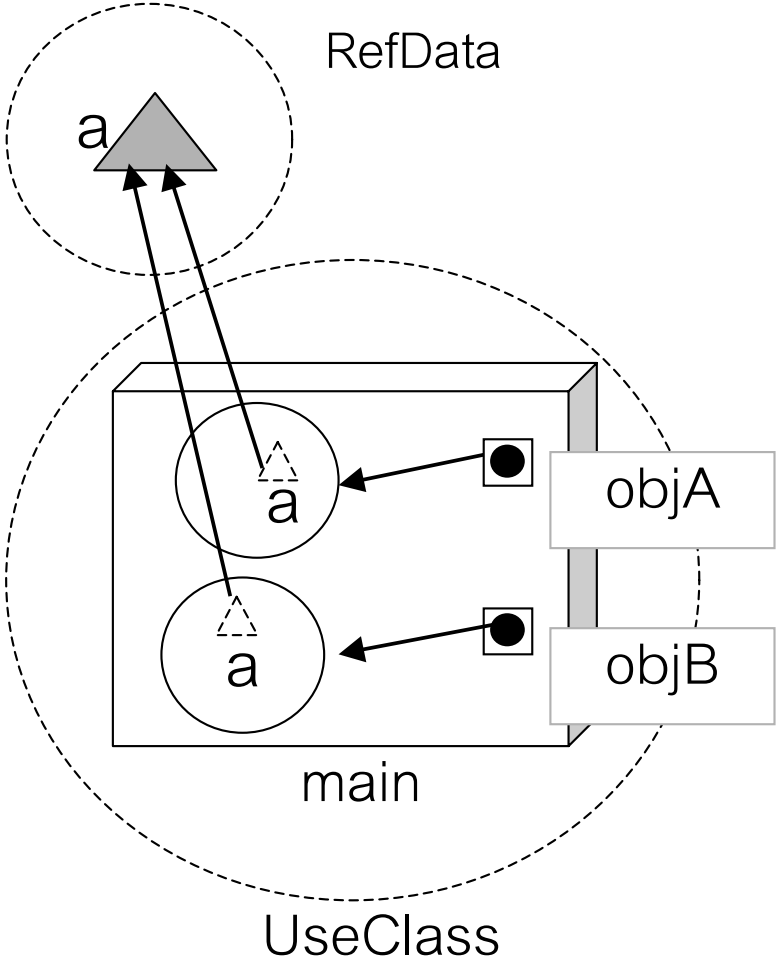


6.2. Method Variables

```
class RefData   {      public static int a;   }
class UseClass  {
    public static void main(String args[])  {
      RefData objA = new RefData( );
      RefData objB = new RefData( );
      objA.a = 10;       objB.a = 20;
      System.out.println(objA.a); System.out.println(objB.a);
    }
}
```

```
java UseClass
20
20
```

6.2. Method Variables

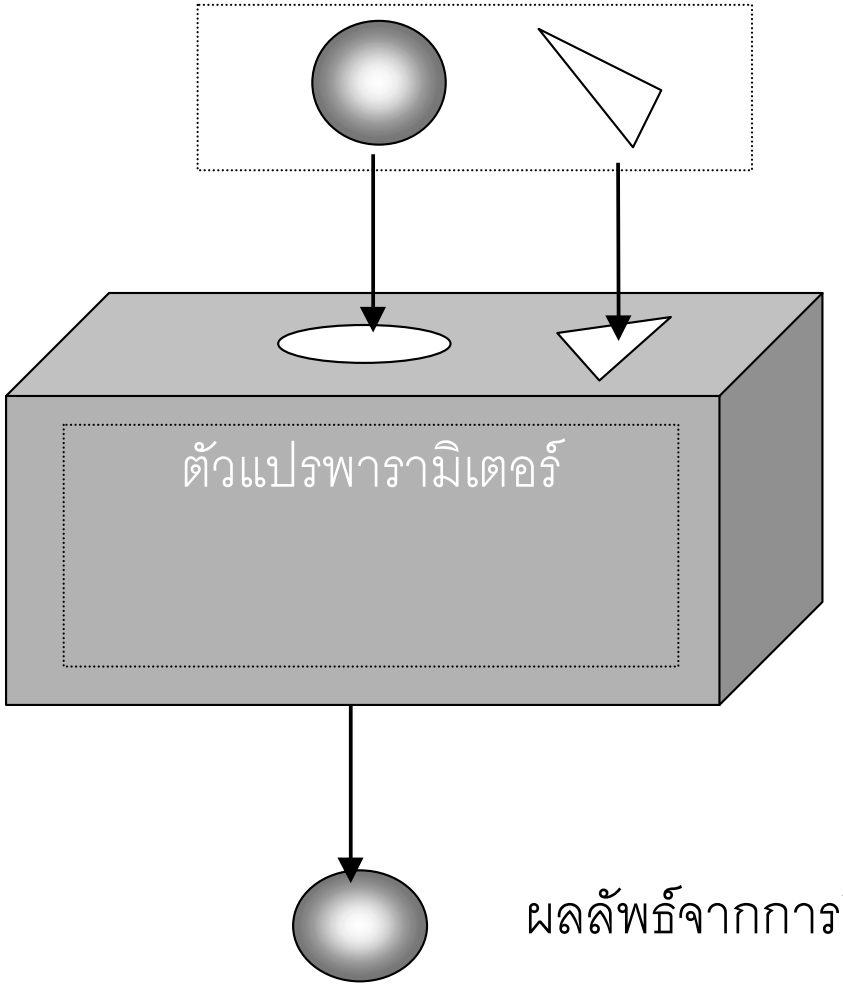


6.3. Parameter Variables

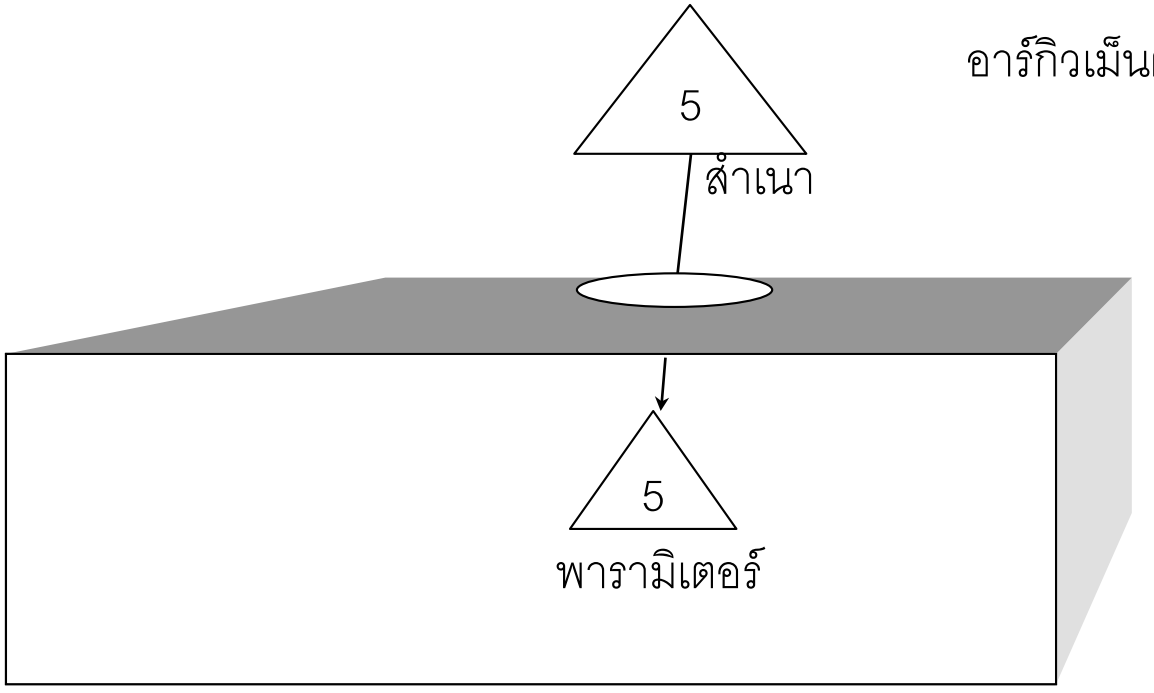
ตัวแปรพารามิเตอร์

- คือตัวแปรที่ใช้ระบุช่องทางรับข้อมูล เพื่อนำข้อมูลเข้าสู่การทำงานของเมธรอด
- สามารถกำหนดโดยใช้ชนิดข้อมูลได้ได้ทั้งแบบ Primitive หรือ Reference
- ถ้าเป็น Primitive การส่งข้อมูลให้เมธรอดเป็นลักษณะสำเนาข้อมูล(Copy Value)
- ถ้าเป็น Class หรือ Array การส่องข้อมูลเป็นแบบอ้างถึง(Reference Value)

6.3. Parameter Variables



6.3. Parameter Variables



Copy Value

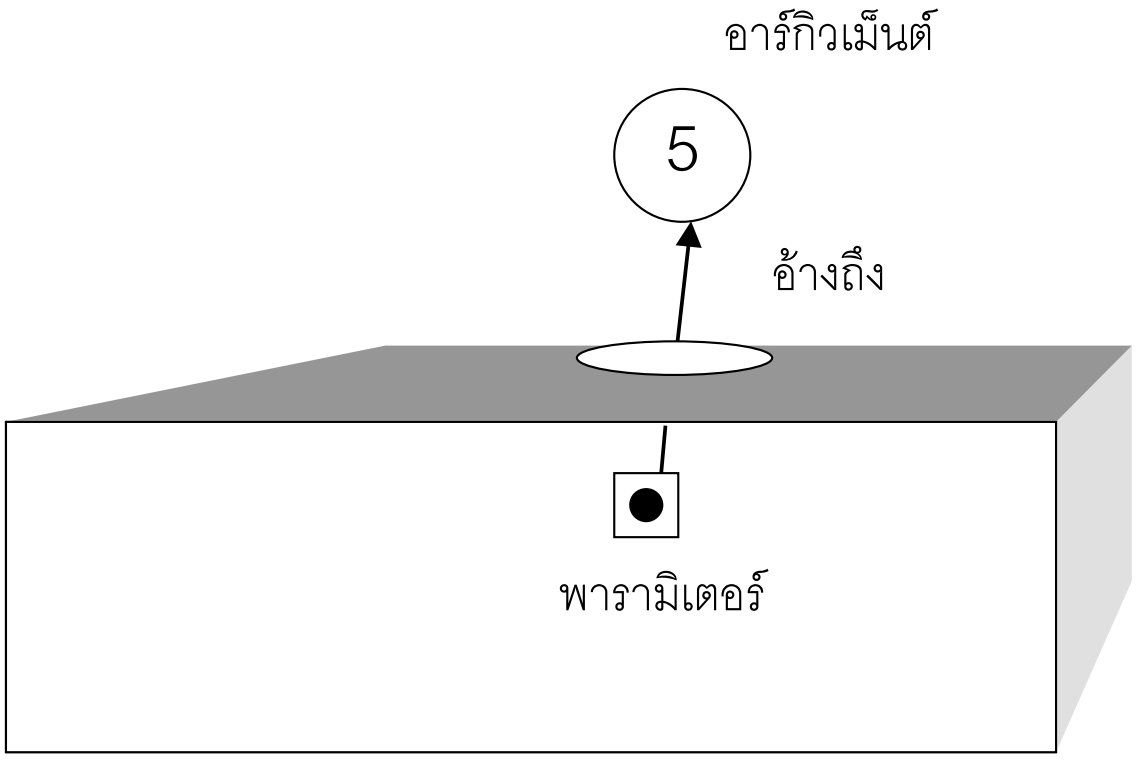
6.3. Parameter Variables

```
class CopyParam {
    public static int computeAdd(int a, int b)
    {           return a + b;                   }
    public static int computeMinus(int a, int b)
    {           return a - b;                   }
    public static void main(String args[])
    {  int number1 = Integer.parseInt(args[0]);
       int number2 = Integer.parseInt(args[1]);
       System.out.println("Result of Your Number");
       System.out.print("\t" + number1 + " + " + number2 + " = ");
       System.out.println(computeAdd(number1, number2));
       System.out.print("\t" + number1 + " - " + number2 + " = ");
       System.out.println(computeMinus(number1, number2));
       }
    }
```

6.3. Parameter Variables

```
java CopyParam 45 12
Result of Your Number
 45 + 12 = 57
 45 - 12 = 33
```

6.3. Parameter Variables



Reference Value

6.3. Parameter Variables

```
class RefParam
{     public static int computeMod(Integer a, Integer b)
      {           return a.intValue( ) % b.intValue( );        }
      public static void main(String args[])
      {           Integer number1 = new Integer(args[0]);
                  Integer number2 = new Integer(args[1]);
                  System.out.print(number1 + " Mod " + number2
                              + " = ");
                  System.out.println(
                              computeMod(number1, number2));
      }
}
```

6.3. Parameter Variables

```
java RefParam 7 2
7 Mod 2 = 1
java RefParam 151 7
151 Mod 7 = 4
```

6.3. Parameter Variables

```
class MyInt
{   int i;          }
class ChangeDataInRefParam
{   public static void Abs1(MyInt a)
      {             a.i = (a.i<0)? a.i * -1: a.i;                 }
      public static int Abs2(MyInt a)
      {             return ( (a.i<0)? a.i * -1: a.i );            }
      public static void main(String args[])
      {             MyInt number1 = new MyInt(); number1.i = -55;
                    MyInt number2 = new MyInt(); number2.i = -184;
                    Abs1(number1);
                    System.out.println("Abs1 " + number1.i);
                    System.out.println("Abs2 " + Abs2(number2));
      }
}
```

6.3. Parameter Variables

```
java ChangeDataInRefParam
Abs1        55
Abs2        184
```