



Chapter 3

02204172

Object and Class



Object

- วัตถุ คือ สิ่งๆ หนึ่งที่มีตัวตน มีข้อมูลและพฤติกรรม
- วงจรชีวิตของวัตถุ
 - สร้างและตั้งค่าเริ่มต้น
 - ใช้งาน
 - ทำลาย



Object (Cont.)

- วัตถุถูกสร้างด้วย operator ที่ชื่อว่า “new”
 - `new String(“ABC”)`
 - `new Date()`
 - `new Integer(1)`
- เมื่อมีการสร้างวัตถุ วัตถุก็จะถูกตั้งค่าเริ่มต้นไปในตัว
- วัตถุทุกตัวจะต้องมีชนิด และชนิดนั้นก็คือ คลาสนั่นเอง
- เราสามารถใช้ตัวแปรมาเก็บวัตถุได้ด้วยการประกาศตัวแปรและกำหนดค่าเป็นวัตถุ
 - `String s = new String(“ABC”);` // s อ้างถึงวัตถุที่เป็นข้อความ “ABC”
 - `Date deadline;` // deadline ไม่ได้อ้างถึงวัตถุใด
 - `deadline = new Date();` // deadline อ้างถึงวันปัจจุบัน



Object (Cont.)

- วัตถุมักถูกใช้งานผ่านตัวแปร
 - การใช้งานวัตถุ คือ การเข้าถึง และการเรียกใช้ method นั้นเอง
- ตัวอย่างการใช้วัตถุใน 4 รูปแบบ
 - 1 ตัวแปร 1 วัตถุ
 - 2 ตัวแปร 1 วัตถุ
 - 2 ตัวแปร 2 วัตถุ
 - 1 ตัวแปร 2 วัตถุ



Object (Cont.)

- 1 ตัวแปร 1 วัตถุ

```
String s = new String ("abc");  
System.out.println(s);  
s.length() == 3;  
s.substring(2);  
s.equals("abc");
```



Object (Cont.)

- 2 ตัวแปร 1 วัตถุ

```
String s1 = new String ("abc");  
String s2;  
s2 = s1;  
s1 = null;  
s2.equals("abc");  
s2.substring(1) == "bc"
```



Object (Cont.)

- 2 ตัวแปร 2 วัตถุ

```
String s1 = new String ("abc");  
String s2 = new String ("abc");  
s2.equals(s1) == true;  
(s1 == s2) == false;
```



Object (Cont.)

- 1 ตัวแปร 2 วัตถุ

```
String s1 = new String ("abc");  
s1 = new String ("cba");
```

“ไม่มีตัวแปรใดเข้าถึงข้อความ "abc" ได้อีก



Object Destruction

- โปรแกรมจะใช้หน่วยความจำเมื่อรัน
 - ตัวแปร
 - วัตถุ
 - โค้ด
- กลไกในการคืนพื้นที่ในหน่วยความจำของโปรแกรมสามารถทำได้ 2 วิธีคือ
 - manual: C++
 - อัตโนมัติ: java
- JRE มี garbage collector ไว้สำหรับคืนพื้นที่ในหน่วยความจำ
- garbage collector ทำงานโดยการวิ่งหาวัตถุที่ไม่เข้าถึงไม่ได้แล้ว แล้วทำการกำจัดมัน (คืนพื้นที่ที่วัตถุนั้นใช้อยู่ให้กับระบบปฏิบัติการ)



Class (Cont.)

- คลาส (ในแง่ทฤษฎีเชิงวัตถุ) คือ รูปแบบหรือพิมพ์เขียว (blueprint) ของวัตถุ โดยเป็นการจัดกลุ่มให้กับวัตถุที่มีคุณสมบัติร่วมกัน
 - โรงงานทำการผลิตวัตถุตามพิมพ์เขียว
 - จะต้องมีการกำหนดคุณสมบัติของวัตถุในกลุ่มนั้น
 - วัตถุถูกผลิตตามแม่พิมพ์



Class (Cont.)

- คลาส (ในเชิงโปรแกรม) คือ โครงสร้างข้อมูลใหม่ ที่เราสามารถสร้างได้ตามต้องการ
 - หน่วยที่มีคุณสมบัติหรือมีชนิดข้อมูลเป็นคลาส (instance of class) เรียกว่า วัตถุ (object)
 - คลาสถือเป็นชนิดข้อมูลที่ซับซ้อน (complex data type)
- Class เปรียบเสมือนตัวแทนของการเขียนโปรแกรมเชิงวัตถุ
- การประกาศคลาสและการใช้วัตถุจึงถือว่าสำคัญมาก



Class (Cont.)

- คลาสแบ่งออกเป็น
 - predefined classes
 - String
 - Date
 - Integer
 - Math
 - API อีกมากมาย
 - user-defined classes



Class (Cont.)

- องค์ประกอบพื้นฐานที่สำคัญของ class ได้แก่
 - Fields
 - Methods
 - Constructor
- สิ่งที่สามารถทำได้กับ class
 - class declaration
 - object instantiation
 - class member access



Class Declaration

```
[<Class_modifier>] class <class-name>
[extends <superclass-name>]
[implements <interface_name_1>, <intf_name_2>,
...]
{
    // body สามารถใส่ได้ทั้ง fields, methods และอื่นๆ
    // ซึ่งเราเรียกสิ่งเหล่านี้ว่าเป็น member ของ class
}
```



Class Declaration (Cont.)

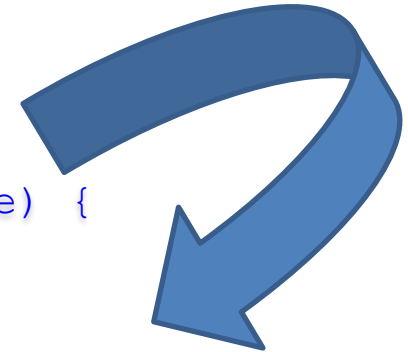
```
public class Person {  
    // attributes  
    public String name;  
    public String surname; } ตัวแปรของ class เรียกว่า field  
    // methods  
    public void show() {  
        System.out.println(name+" "+surname)  
    }  
}
```



Constructor

- ในคลาสหนึ่งๆ จะมี method พิเศษที่เรียกว่า constructor ซึ่งจะต้องมีชื่อเดียวกับชื่อของคลาส และไม่มี return type ใน header
- เป็นส่วนที่ใช้กำหนดค่าเริ่มต้นให้วัตถุ

```
public class Person {  
    // attributes  
    ...  
    // constructor  
    public Person(String name, String surname) {  
        this.name = name;  
        this.surname = surname;  
    }  
    public Person() {  
        name = "John";  
        surname = "Smith";  
    }  
    // methods  
    ...  
}
```





Constructor (Cont.)

- ประกาศตัวแปรที่มีชนิดเป็น Person ค่าของ attributes ของวัตถุแต่ละอันคืออะไรบ้าง



- Person p1 = new Person();
- Person p2 = new Person("sue", "lilly");
- Person p3;
- Person p4 = p1;





Constructor (Cont.)

- Default constructor คือ constructor ที่ไม่มี parameter ใดๆ
- หากคลาสใดไม่มีการประกาศ constructor ไว้โปรแกรมจะสร้าง default constructor ให้เอง
- หากคลาสมีการประกาศ constructor ไว้โปรแกรมจะสร้างไม่สร้าง default constructor ให้เอง
- attribute ที่เป็นชนิดพื้นฐานจะมีค่าเริ่มต้นตามชนิด เช่น int จะเป็น 0
- attribute ที่เป็นวัตถุจะมีค่าเริ่มต้นเป็น null



Constructor (Cont.)

- ข้อใดประกาศคลาสและ constructor ได้ถูกต้อง

```
class Box{  
    ...  
    public MyBox() { ... }  
    ...  
}
```

```
class Box{  
    int width;  
    int height;  
    ...  
}
```

```
class Dog{  
    ...  
    public Dog() { ... }  
    ...  
}
```

```
class Dog{  
    ...  
    public Dog() { ... }  
    public Dog(int age) {...}  
    ...  
}
```

```
class Dog{  
    ...  
    void Dog() { ... }  
    ...  
}
```



Practice

- จงประกาศ constructor ของคลาส Point
 - default constructor โดยกำหนดให้ $x=0$ และ $y=0$
 - constructor ที่ใส่ x และ y เป็นพารามิเตอร์
- จงประกาศ constructor ของคลาส Triangle ที่ใส่ความสูงและฐานเป็นพารามิเตอร์



Class Member

- Member คือ ทุกองค์ประกอบที่อยู่ภายใน class
 - Fields → ตัวแปร
 - Constants
 - Methods
 - Constructors

```
public class Person {  
    // attributes  
    public String name;  
    public String surname;  
    // methods  
    public void show() {  
        System.out.println(name+" "+surname)  
    }  
}
```



Practice

- จงประกาศตัวแปร และสร้างวัตถุต่อไปนี้
- คนที่มีชื่อเป็น “lilly” และนามสกุลเป็น “Devaux”
- จุดที่อยู่ ณ จุดกำเนิด
- จุด $(x,y) = (10,3)$
- สามเหลี่ยมที่มีความสูงเป็น 5 และฐานเป็น 4



Class Members Access

- การเข้าถึงสมาชิกของ class มี 2 แบบ คือ

- ภายใน class เอง: เรียกใช้ได้เลย หรือ

```
this.<member-name>
```

- ภายนอก class: โดยสมาชิกนั้นจะต้องเป็น public หรือ internal

```
< var-name >.<member-name>
```



Class Members Access (Cont.)

```
public class Person {  
    // attributes  
    public String name;  
    public String surname;  
    // methods  
    public void show(String name) {  
        System.out.println(this.name+" "+surname)  
    }  
}
```




Class Members Access (Cont.)

```
public class Person {  
    // attributes  
    public String name;  
    public String surname;  
    // methods  
  
    public Person() {  
        this.name = "";  
        this.surname = "" ;  
    }  
}
```



Class Members Access (Cont.)

```
public class Person {  
    // attributes  
    public String name;  
    public String surname;  
    // methods  
  
    public Person() {  
        this("", "");  
    }  
  
    public Person(String name, String surname) {  
        this.name = name;  
        this.surname = surname;  
    }  
}
```



Class Members Access (Cont.)

- การเรียกใช้ attribute
 - รูปแบบ: `objectName.attributeName`
 - ข้อควรปฏิบัติ: ถึงแม้ว่าจะสามารถเรียกใช้ attribute ได้โดยตรง แต่โดยปกติแล้ว การเขียนคลาสที่ดี จะต้องไม่ให้วัตถุใดเข้าถึง attribute ได้โดยตรง แต่ต้องเรียกผ่าน method ดังนั้น คำสั่งนี้ จึงแทบไม่มีการใช้งานและ**ไม่ควรใช้อย่างยิ่ง**
- การเรียกใช้ method
 - รูปแบบ: `objectName.methodName([arguments]);`
 - ตัวอย่าง:
 - `person.setName("susan");`
 - `System.out.println(person.getName());`



Class Members Access (Cont.)

```
public class ClassTest {  
    public static void main(String[] args) {  
        Person person1 = new Person("Amie", "Roy");  
        Person person2 = new Person();  
        person2.name = "Rod";  
        person2.surname = "Wales";  
        person1.show();  
        person2.show();  
    }  
}
```



Practice

- จงประกาศ method ต่อไปนี้สำหรับคลาส Point
 - `setX(int x)` สำหรับเปลี่ยนค่าของ field `x` ในคลาส
 - `setY(int y)` สำหรับเปลี่ยนค่าของ field `y` ในคลาส
- เมื่อประกาศแล้ว ให้สร้างจุด 2 จุดใน main โดยจุดแรกสร้างโดย default constructor และจุดที่สองให้กำหนดเป็น (4,5) หลังจากนั้นเปลี่ยนค่า `x` ของจุดแรกเป็น 2 และเปลี่ยนค่า `y` ของจุดที่ 2 เป็น 0



Practice (Cont.)

- จงประกาศ method ชื่อ `getArea()` ของคลาส `Triangle` เพื่อคำนวณหาพื้นที่สามเหลี่ยม
- สร้างสามเหลี่ยมใน `main()` โดยให้มีความสูง 7 ฐาน 5 แล้วคำนวณหาพื้นที่สามเหลี่ยม พร้อมทั้งแสดงออกมาทางหน้าจอ



Data types

- ชนิดข้อมูลแบ่งออกเป็น 2 ประเภท คือ
 - value types: ตัวแปรที่มีชนิดแบบ value type จะเก็บข้อมูลในชนิดนั้นโดยตรง
 - ชนิดพื้นฐาน: int, float, double, char

number

15

letter

'A'



Data types (Cont.)

- reference types: ตัวแปรที่มีชนิดแบบ reference type จะเก็บ reference ไปยังข้อมูลจริง แทนที่จะเก็บข้อมูลจริงลงไป
 - string แต่ใช้งานได้เหมือนกับ value type
 - array
 - class

