# บทที่ 4 : หลักการเชิงอ็อบเจกต์

## วัตถุประสงค์

- เพื่อเรียนรู้การประกาศคลาส และการสร้างอ็อบเจกต์
- เพื่อเรียนรู้การประกาศคุณลักษณะและเมธอด
- เพื่อเรียนรู้การเรียกใช้งานเมธอด
- เพื่อให้เข้าใจหลักการเขียนโปรแกรมเชิงอ็อบเจกต์
- เพื่อให้เข้าใจการเขียนโปรแกรมที่มีการสร้าง constructor แบบ overloaded
- เพื่อให้เข้าใจความหมายของเมธอดแบบ overloaded และเมธอดแบบ overridden
- เพื่อให้เข้าใจความหมายของคลาสชนิด `abstract` และอินเตอร์เฟส (Interface)

## แบบฝึกหัดเชิงปฏิบัติการ

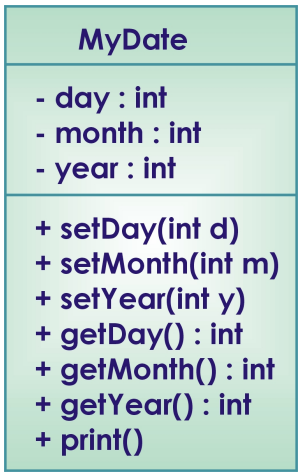
**แบบฝึกหัดที่ 1** การเขียนโปรแกรมโดยใช้หลักการของการห่อหุ้ม

### วัตถุประสงค์

1) เพื่อให้เข้าใจหลักการการเขียนโปรแกรมเชิงอ็อบเจกต์โดยการใช้หลักการของการห่อหุ้ม

### ขั้นตอนการปฏิบัติการ

1) จงเขียนคลาส `MyDate` ที่มีไดอะแกรมของคลาสดังแสดงในรูปที่ 4.1




**รูปที่ 4.1** ไดอะแกรมของคลาส `MyDate`

โดยมีขั้นตอนดังนี้

(a) กำหนดคลาสที่ชื่อ `MyDate` ในไฟล์ที่ชื่อ `MyDate.java` ซึ่งมีรูปแบบดังนี้

```
public class MyDate {
        ...
}
```

(b) กำหนดคุณลักษณะของคลาส `MyDate` คือ `day`, `month` และ `year` ให้มีชนิดข้อมูลเป็นแบบ `int` และมี access modifier เป็น `private`

(c) กำหนดเมธอดแบบ accessor เพื่อ set และ get ค่าของคุณลักษณะทั้งสาม

(d) กำหนดเมธอด `print()` เพื่อแสดงค่าวันเดือนปี ออกทางจอภาพโดยมีรูปแบบเป็น `dd/mm/yyyy`

2) เขียนคลาสที่ชื่อ `MyMain` โดยกำหนดให้มีเมธอด `main()` อยู่ภายในคลาส โดยเขียนคำสั่งภายในเมธอด `main()` ดังนี้
   (a) ประกาศและสร้างอ็อบเจกต์ของคลาส `MyDate` ที่ชื่อ `d1`
   (b) เรียกใช้เมธอดแบบ setter เพื่อกำหนดค่าของคุณลักษณะวันเดือนปีของ อ็อบเจกต์ `d1` โดยให้ `day` = 16, `month` = 12 และ `year` = 2002
   (c) เรียกใช้เมธอด `print()` เพื่อแสดงค่าวันเดือนปี 16/12/2002 ของ อ็อบเจกต์ `d1` ออกทางจอภาพ
   (d) เรียกใช้เมธอดแบบ setter เพื่อกำหนดค่าของคุณลักษณะวันเดือนปีของ อ็อบเจกต์ `d1` โดยให้ `day` = 29, `month` = 2 และ `year` = 2002
   (e) เรียกใช้เมธอด `print()` เพื่อแสดงค่าวันเดือนปี 29/2/2002 ของอ็อบเจกต์ `d1` ออกทางจอภาพ
3) คอมไพล์และรันโปรแกรมที่พัฒนาขึ้น
4) จงปรับปรุงคำสั่งในเมธอดแบบ setter เพื่อให้สามารถรับค่าของคุณลักษณะแต่ละตัวในรูปแบบที่ถูกต้อง เช่นคุณลักษณะ `day` จะต้องมีค่าได้ไม่เกิน 31 เป็นต้น

**แบบฝึกหัดที่ 2** การเขียน constructor และเมธอดแบบ overloaded

**วัตถุประสงค์**

1) เพื่อให้เข้าใจการเขียนโปรแกรมโดยใช้หลักการของการห่อหุ้ม
2) เพื่อให้เข้าใจการเขียน constructor ของคลาส
3) เพื่อให้เข้าใจการเขียน constructor แบบ overloaded
4) เพื่อให้เข้าใจการสร้างอ็อบเจกต์โดยใช้คำสั่ง `new`

**ขั้นตอนการปฏิบัติการ**

1) โปรแกรมที่ 5.1 แสดงโครงสร้างของคลาส `Rectangle` จงเขียนคำสั่งในเมธอดต่างๆ ต่อไปนี้ให้สมบูรณ์

**โปรแกรมที่ 4.1** โครงสร้างของคลาส `Rectangle`

```
public class Rectangle {
      private double width = 1;
      private double height = 1;
      private static String color = "white";

      public Rectangle() {
      }
      public Rectangle(double w, double h, String c) {
      }

      public void setWidth(double width) {
      }
      public double getWidth() {
      }
      public void setHeight(double height) {
      }
      public double getHeight() {
      }

      public void setColor(String color) {
      }
      public String getColor() {
      }
      public double findArea() {
      }
}
```

a. เขียน constructor ที่มีรูปแบบ
   `public Rectangle(double w, double h, String c) { }`
   เพื่อกำหนดค่าเริ่มต้นให้กับคุณลักษณะที่ชื่อ `width`, `height` และ `color`

b. เขียนคำสั่งในเมธอดที่เป็น getter และ setter ทั้งหมด

c. เขียนคำสั่งในเมธอด `findArea()` เพื่อคำนวณหาพื้นที่ของสี่เหลี่ยมที่มีค่าเป็น `width * height` และส่งค่าดังกล่าวคืนมาโดยใช้คำสั่ง `return`

2) กำหนดคลาส `TestRectangle` และกำหนดให้มีเมธอด `main()` ภายในคลาสนี้ เพื่อใช้ในการทดสอบและสร้างอ็อบเจกต์ของคลาส `Rectangle`

3) เขียนคำสั่งประกาศและสร้างอ็อบเจกต์ของคลาส `Rectangle` ขึ้นมาสองอ็อบเจกต์

   - อ็อบเจกต์ที่หนึ่งชื่อ `rect1` โดยใช้ constructor แบบ default ที่มี คำสั่งดังนี้
     `rect1 = new Rectangle()`
   - อ็อบเจกต์ที่สองชื่อ `rect2` โดยใช้ constructor แบบที่สองเพื่อกำหนดค่าของคุณลักษณะที่ชื่อ `width`, `height` และ `color` ให้เป็น `10.5`, `8` และ `red` ตามลำดับ

4) เขียนคำสั่งโดยใช้เมธอดแบบ setter เพื่อกำหนดค่าคุณลักษณะของอ็อบเจกต์ `rect1` ที่ชื่อ `width`, `height` และ `color` ให้เป็น `12`, `5.5` และ `yellow` ตามลำดับ

5) เรียกใช้เมธอดแบบ getter เพื่อแสดงค่าคุณลักษณะต่างๆของอ็อบเจกต์ทั้งสอง

6) เรียกใช้เมธอด `findArea()` เพื่อคำนวณหาและแสดงพื้นที่ของอ็อบเจกต์ทั้งสอง

**แบบฝึกหัดที่ 3** การเขียนโปรแกรมบัญชี

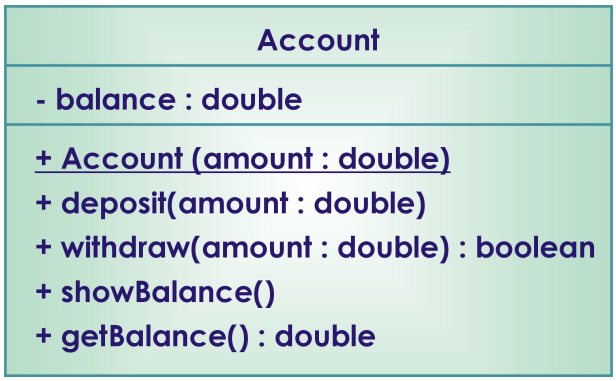
**วัตถุประสงค์**

1) เพื่อให้เข้าใจการพัฒนาโปรแกรมโดยใช้หลักการเชิงอ็อบเจกต์

2) เพื่อให้เข้าใจการพัฒนาโปรแกรมจากไดอะแกรมของคลาสที่กำหนดมาให้

**ขั้นตอนการปฏิบัติการ**

1) จงเขียนคลาสที่ชื่อ `Account` ซึ่งมีคลาสไดอะแกรมดังแสดงในรูปที่ 4.2




**รูปที่ 4.2** ไดอะแกรมของคลาส `Account`

โดยมีขั้นตอนดังนี้

(a) เขียน constructor ที่มี argument ในการรับยอดเงินเปิดบัญชี

(b) เขียน constructor แบบ default ที่ไม่มีคำสั่งใดภายใน

(c) กำหนดคุณลักษณะ `balance` ที่เป็นชนิดข้อมูลแบบ `double` และมี access modifier เป็น `private`

(d) เขียนเมธอดที่ชื่อ `deposit()` เพื่อใช้ในการฝากเงินโดยมี argument ที่ใช้ในการรับจำนวนเงินที่ต้องการฝาก

(e) เขียนเมธอดที่ชื่อ `withdraw()` เพื่อใช้ในการถอนเงินโดยมี argument ที่ใช้ในการรับจำนวนเงินที่ต้องการถอน เมธอดนี้จะส่งค่า `true` กลับมาถ้าสามารถถอนเงินได้โดยมียอดเงินในบัญชีมากกว่าจำนวนที่ต้องการถอน และจะส่งค่า `false` กลับมาถ้ายอดเงินที่ต้องการถอนมากกว่ายอดเงินในบัญชี โดยจะไม่ทำให้ยอดเงินในบัญชีเปลี่ยนแปลง

(f) เขียนเมธอด `getBalance()` ที่ส่งยอดเงินในบัญชีกลับคืนมา

(g) เขียนเมธอด `showBalance()` เพื่อพิมพ์ยอดเงินในบัญชีออกทางจอภาพโดยใช้คำสั่ง `System.out.println()`

2) เขียนคลาสที่ชื่อ `Teller1` ซึ่งจะเป็นคลาสที่ใช้ในการสร้างอ็อบเจกต์ของคลาส `Account` และทดสอบการเรียกใช้เมธอดต่างๆโดยมีขั้นตอนดังนี้

(a) กำหนดเมธอด `main()` ภายในคลาส `Teller`

(b) ประกาศและสร้างอ็อบเจกต์ที่ชื่อ `acc` โดยกำหนดให้มีเงินเปิดบัญชีเป็น `5,000` บาท

(c) เรียกใช้เมธอด `deposit()` เพื่อฝากเงินจำนวน `2,000` บาท

(d) เรียกใช้เมธอด `showBalance()` เพื่อแสดงยอดเงินคงเหลือ

(e) เรียกใช้เมธอด `withdraw()` เพื่อถอนเงินจำนวน `4,000` บาท

(f) เรียกใช้เมธอด `showBalance()` เพื่อแสดงยอดเงินคงเหลือ

(g) เรียกใช้เมธอด `withdraw()` เพื่อถอนเงินจำนวน `8,000` บาท

(h) เรียกใช้เมธอด `showBalance()` เพื่อแสดงยอดเงินคงเหลือ

(i) เรียกใช้เมธอด `withdraw()` เพื่อถอนเงินจำนวน `2,000` บาท

(j) เรียกใช้เมธอด `showBalance()` เพื่อแสดงยอดเงินคงเหลือ

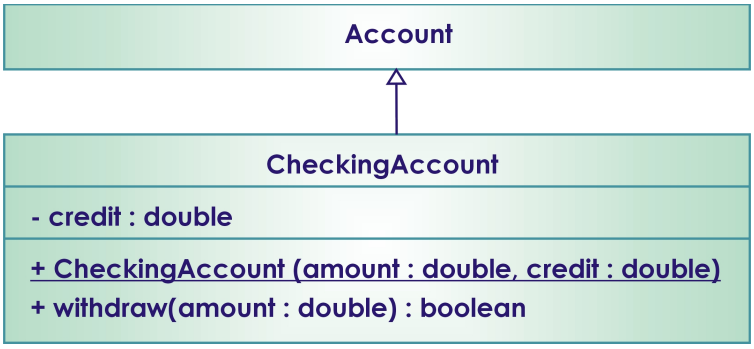
**แบบฝึกหัดที่ 4** การเขียนโปรแกรมบัญชีเช็ค

**วัตถุประสงค์**

1) เพื่อให้เข้าใจการเขียนโปรแกรมโดยใช้หลักการของการสืบทอด
2) เพื่อให้เข้าใจการเขียนเมธอดแบบ overridden
3) เพื่อให้เข้าใจหลักการการกำหนดอ็อบเจกต์แบบ Dynamic Binding

**ขั้นตอนการปฏิบัติการ**

1) จงเขียนคลาสที่ชื่อ `CheckingAccount` ซึ่งมีคลาสไดอะแกรมดังแสดงในรูปที่6.2 โดยมีขั้นตอนดังนี้




**รูปที่ 5.2** ไดอะแกรมของคลาส `CheckingAccount`

(a) เปลี่ยนคุณลักษณะของคลาส `Account` ที่ชื่อ `balance` ให้มี access modifier เป็น `protected`

(b) กำหนดให้คลาส `CheckingAccount` สืบทอดมาจากคลาส `Account`

(c) กำหนดคุณลักษณะ `credit` ที่เป็นชนิดข้อมูลแบบ `double` และมี access modifier เป็น `private`

(d) เขียนเมธอดแบบ overridden ที่ชื่อ `withdraw()` โดยจะอนุญาตให้ถอนเงินเกินบัญชีได้ ถ้ายอดที่ถอนอยู่ภายในวงเงินเกินบัญชีที่ยังเหลืออยู่ `(balance + credit)`

(e) เพิ่มเมธอดที่ชื่อ `showCredit()` เพื่อพิมพ์ยอดของ `credit` ออกทางจอภาพ

## คำถามทบทวน

1. modifier ใดบ้างที่ถือว่าเป็นประเภท access modifier
2. จงอธิบายระดับการเข้าถึงของ access modifier แต่ละตัว
3. ตัวแปรและเมธอดประเภทใดที่ควรจะมี modifier เป็น `static`
4. จงอธิบายความหมายของคำว่า superclass และ subclass
5. คลาสทุกคลาสที่ใช้ในภาษาจาวาจะถือว่าสืบทอดมาจากคลาสใด
6. จงอธิบายว่าคลาสที่จะสืบทอดกันได้ควรมีความสัมพันธ์ต่อกันอย่างไร

7. จงอธิบายความแตกต่างระหว่างเมธอดแบบ overloaded และเมธอดแบบ overridden
8. อะไรทำให้ default constructor แตกต่างจาก constructor ทั่วๆไป
9. อินเตอร์เฟสแตกต่างจากคลาสแบบ `abstract` อย่างไร
10. ข้อใดเป็นคำสั่งภาษาจาวาที่ถูกต้อง

```
a. y++;
b. public Date d;
c. public class String2 extends String{}
d. public class Nothing{}
e. public void static method1() {}
f. public abstract final method1();
```

11. ผลลัพธ์ของโปรแกรมต่อไปนี้คืออะไร

a)

```
public class Ex6_1_10a {
   public static void main(String args[]) {
      int x;
      System.out.println(x);
   }
}
```

b)

```
public class Ex6_1_10b {
   int x;
   public void method1() {
      System.out.println(x);
   }
   public static void main(String args[]) {
      Ex6_1_10b obj = new Ex6_1_10b();
      obj.method1();
   }
}
```

c)

```
public class Ex6_1_10c{
   int x;
   public static void main(String args[]) {
      System.out.println(x);
   }
}
```

12. จงเขียนคำสั่งเพื่อเรียกเมธอด `init()` เมธอด main ภายในคลาสต่อไปนี้

```
public class Sample {

   public void init() {
      System.out.println("Please call me");
   }

   public static void main(String args[]) {

   }
}
```

13. จงเขียนเมธอดโดยใช้คำสั่ง `Math.random()` เพื่อที่จะส่งค่าจำนวนเต็มที่สุ่มมาระหว่างเลข -100 ถึง 100

## แบบฝึกหัดทบทวน

1. จงเขียนคลาสที่ชื่อ `ComplexNumber` ซึ่งมีคุณลักษณะสองตัวคือ `re` และ `im` เพื่อเก็บค่าส่วนจริงและจินตภาพ และมีเมธอดที่ใช้ในการบวก ลบ และ คูณ อ็อบเจกต์ ชนิด `ComplexNumber` สองตัว
2. จงเขียนเมธอดแบบ overridden ที่ชื่อ `toString()` และ `equals()` เพื่อแปลงอ็อบเจกต์ชนิด `ComplexNumber` ให้เป็นข้อความ และเปรียบเทียบกับ อ็อบเจกต์ชนิด `ComplexNumber` ตัวอื่น
3. จากคลาส `Employee` ในแบบฝึกหัดทบทวนข้อที่ 1 ของบทที่ 4 จงเขียนคลาสที่ชื่อ `Manager` ซึ่งสืบทอดมาจากคลาส `Employee` และมีคุณลักษณะที่ชื่อ `department` เพิ่มขึ้นมา และ override เมธอด `showDetails()` และกำหนดเมธอด `main()` ในการสร้างอ็อบเจกต์ชนิด `Manager`
4. จงเขียนโปรแกรมข้อ 3 ใหม่ โดยกำหนดให้ `Employee` เป็นอินเตอร์เฟส และกำหนดคุณลักษณะทั้งหมดในคลาส `Manager`

## คำถามทบทวน

1. ให้ลองยกตัวอย่างของคุณลักษณะและเมธอดต่างๆ ที่อ็อบเจกต์ต่อไปนี้ควรจะมี
    a. เสื้อ
    b. ผู้จัดการ
    c. บัญชีธนาคาร
2. คลาสแตกต่างจากอ็อบเจกต์อย่างไร
3. อะไรคือความแตกต่างระหว่างคุณลักษณะของอ็อบเจกต์และคุณลักษณะของคลาส
4. จงบอกหน้าที่ของเมธอด
5. โปรแกรมจาวาประยุกต์จะเริ่มต้นการทำงานในคลาสที่มีเมธอดใด
6. จงบอกถึงความแตกต่างระหว่างส่วนที่เป็น interface และส่วนที่เป็น implementation ของอ็อบเจกต์
7. ในภาษาจาวาคีย์เวิร์ดใดถูกใช้เมื่อต้องการระบุการสืบทอด
8. อะไรเป็นข้อดีของการมีได้หลายรูปแบบ
9. จงอธิบายความหมายและหน้าที่ของ UML
10. ขั้นตอนใดเป็นขั้นตอนที่นักพัฒนาโปรแกรมที่ดีควรจะให้ความสำคัญมากที่สุด

## แบบฝึกหัดทบทวน

1. จงเขียนคลาสที่ชื่อ `Employee` ซึ่งมีคุณลักษณะสามตัวคือ `name, position` และ `salary` โดยเขียนโปรแกรมที่ใช้หลักการของการห่อหุ้ม กำหนดเมธอดที่ชื่อ `calSalary()` ซึ่งรับข้อมูลคือจำนวนชั่วโมงการทำงานชนิด `int` และอัตราค่าจ้างต่อชั่วโมงชนิด `double` เข้ามาและส่งผลคูณของค่าทั้งสองกลับไปเป็นชนิด`double` จากนั้นให้กำหนดเมธอด `showDetails()` เพื่อแสดงรายละเอียดของคุณลักษณะทั้งสาม และกำหนดเมธอด `main()` ในการสร้างอ็อบเจกต์ของคลาส `Employee` และทดลองเรียกใช้เมธอดต่างๆ
2. เขียนข้อกำหนดของบัญชีเงินฝากธนาคาร (Account) โดยระบุรายชื่อคุณลักษณะและเมธอดที่ควรมี และเขียนเค้าโครงร่างของคลาส `Account` โดยใช้หลักการของการห่อหุ้ม