

## แบบฝึกปฏิบัติ ครั้งที่ 4

เรื่อง การเขียนโปรแกรมเชิงวัตถุเบื้องต้น

- วัตถุประสงค์
1. เพื่อฝึกฝนกระบวนการพัฒนาโปรแกรมเชิงวัตถุ
  2. เพื่อฝึกฝนการสร้างคลาส วัตถุ และทริบิวท์ และ เมธอด
  3. เพื่อฝึกฝนการโปรแกรมให้วัตถุสามารถสื่อสารร่วมกัน

1. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MyAdd cm = new MyAdd();

        System.out.println("Please insert number : ");
        int x = sc.nextInt();
        int num = cm.AddTwo(x);
        System.out.println("The result#1 is "+num);

        num = cm.AddTwo(x+5);
        System.out.println("The result#2 is "+num);

        num = cm.AddTwo(x*3+2);
        System.out.println("The result#3 is "+num);
    }
}

public class MyAdd {
    public int AddTwo ( int a ) {
        return (a + 2);
    }
}
```

- 1.1. ผลลัพธ์ที่ได้จากการรันโปรแกรมข้างต้น เมื่อผู้ใช้กรอก 10 ผ่านทางคีย์บอร์ด

Please insert number :

10

The result#1 is 12

The result#2 is 17

The result#3 is 34

- 1.2. ผลลัพธ์ที่ได้จากการรันโปรแกรมข้างต้น เมื่อผู้ใช้กรอก -5 ผ่านทางคีย์บอร์ด

Please insert number :

-5

The result#1 is -3

The result#2 is 2

The result#3 is -11

2. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

```

import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MyAdd2 obj = new MyAdd2();

        System.out.println("Please insert number1 : ");
        int x = sc.nextInt();
        System.out.println("Please insert number2 : ");
        int num = sc.nextInt();

        int result = obj.AddTwo(x);
        System.out.println("The result#1 is "+ result);

        result = obj.AddNum (x+5, num);
        System.out.println("The result#2 is "+ result);

        result = obj.AddTwo(x*3+2);
        System.out.println("The result#3 is "+ result);
    }
}

public class MyAdd2 {
    public int AddTwo ( int a ) {
        return AddNum ( a, 2 ) ;
    }
    public int AddNum ( int a, int num ){
        return (a + num) ;
    }
}

```

2.1. ผลลัพธ์ของโปรแกรม เมื่อผู้ใช้กรอก x เป็น 10 และ num เป็น 5 ผ่านทางคีย์บอร์ด

Please insert number1 :

10

Please insert number2 :

5

The result#1 is 12

The result#2 is 20

The result#3 is 34

2.2. ผลลัพธ์ของโปรแกรม เมื่อผู้ใช้กรอก x เป็น -5 และ num เป็น 10 ผ่านทางคีย์บอร์ด

Please insert number1 :

-5

Please insert number2 :

10

The result#1 is -3

The result#2 is 10

The result#3 is -11

### 3. ให้นักศึกษาสร้างคลาส Student ตามคลาสไดอะแกรมต่อไปนี้

Student	
+ name	: String
+ mScore	: double
+ fScore	: double
+ showGrade ()	: void

โดยกำหนดให้คลาส Student มีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างแอ็ททริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- ให้นักศึกษาสร้างเมธอด showGrade () โดยเริ่มจากการคำนวนคะแนน (s) จากสมการต่อไปนี้

$$s = (mScore \times 0.4) + (fScore \times 0.4) + 20$$

โดยที่ mScore และ fScore เป็นค่ามาจากการแอ็ททริบิวท์ จากนั้น นำค่าคะแนนข้างต้นมาทำการกำหนดเกรด (grade) ดังข้อกำหนดต่อไปนี้

ช่วงคะแนน	น้อยกว่า 50	$50 \leq s < 60$	$60 \leq s < 70$	$70 \leq s < 80$	$80 \leq s$
เกรด	F	D	C	B	A

แล้วจึงแสดงข้อความต่อไปนี้

Your grade is [grade].

หมายเหตุ ค่าของ mScore และ fScore อยู่ในช่วง 0 ถึง 100 หากนั้น นักศึกษาไม่ต้องสร้างเงื่อนไขมาตรวจสอบ

โค้ดสำหรับทดสอบความถูกต้องของคลาส Student

```
public class Main {
    public static void main(String[] args) {
        Student s = new Student();
        s.mScore = 80;
        s.fScore = 80;
        s.showGrade();
    }
}
```

ตัวอย่างผลลัพธ์

Your grade is A.

4. ให้นักศึกษาสร้างคลาสตามไดอะแกรม Fraction ตามที่กำหนดให้ต่อไปนี้

<b>Fraction</b>	$\frac{topN}{btmN}$
+ topN : int	
+ btmN : int	
+ toFraction() : String	
+ toFloat() : String	
+ addFraction(Fraction f) : void	

โดยกำหนดให้คลาส Fraction มีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างแอ็พทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- เมธอด **toFraction()** จะ return ข้อความให้อยู่ในรูปแบบเศษส่วน ดังต่อไปนี้ “[ค่า topN]/[ค่า btmN]”
- เมธอด **toFloat()** จะ return ข้อความให้อยู่ในรูปแบบทศนิยม
- เมธอด **addFraction(Fraction f)** จะดำเนินการบวกเศษส่วนจากตัวแปร f เข้าไปในเศษส่วนของตัวเอง โดยกำหนดให้  $topN_{self}$  และ  $btmN_{self}$  คือตัวเศษและตัวส่วนของตัวเอง ขณะที่  $topN_F$  และ  $btmN_F$  คือ ตัวเศษและตัวส่วนของตัวแปร F
  - กรณีที่ 1 ถ้าส่วน (btmN) ของตัวเองกับตัวแปร f เท่ากันแล้ว ให้นำค่าของเศษ (topN) ของตัวเอง กับตัวแปร f บวกกัน ดังสมการต่อไปนี้
 
$$topN_{self} = topN_{self} + topN_F \text{ และ } btmN_{self} = btmN_{self}$$
  - กรณีที่ 2 ถ้าส่วน (btmN) ของตัวเองกับตัวแปร f ไม่เท่ากันแล้ว ให้อาร์กิวเมนต์ต่อไปนี้ในการคำนวณหาค่า  $topN_{self}$  และ  $btmN_{self}$ 

$$topN_{self} = topN_{self} \times btmN_F + topN_F \times btmN_{self} \text{ และ } btmN_{self} = btmN_{self} \times btmN_F$$

โค้ดสำหรับทดสอบความถูกต้องของคลาส **Fraction**

```
public class Main {
    public static void main(String[] args) {
        Fraction f1 = new Fraction();
        f1.topN = 2;
        f1.btmN = 5;
        Fraction f2 = new Fraction();
        f2.topN = 3;
        f2.btmN = 7;
        System.out.println("before " + f1.toFraction());
        System.out.println("before " + f1.toFloat());
        f1.addFraction(f2);
        System.out.println("after " + f1.toFraction());
        System.out.println("after " + f1.toFloat());
    }
}
```

ตัวอย่างผลลัพธ์

```
before 2/5
before 0.4
after 29/35
after 0.8285714285714286
```

5. ให้นักศึกษาสร้างเมธอดลงในคลาส **Fraction** เพิ่มเติม ดังนี้

5.1. เมธอด **myEquals()** เพื่อเปรียบเทียบว่าเศษส่วนทั้ง 2 มีค่าเท่ากันหรือไม่

```
public boolean myEquals(Fraction x) {
    return this.toFloat().equals(f.toFloat());
}
```

```
int a = topN;
int b = btmN;
while (b != 0) {
    int x = a % b;
    a = b;
    b = x;
}
int gcd = a;
topN = topN / gcd;
btmN = btmN / gcd;
```

5.2. เมธอด **LowestTermFrac()** เพื่อทำให้เศษส่วนตัวดังกล่าวเป็นเศษส่วนอย่างต่ำ

```
public void LowestTermFrac() {
    .....
}
```

โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาส **Fraction** ที่นักศึกษาได้พัฒนาขึ้นมา

```
public class Main {
    public static void main(String[] args) {
        Fraction f1 = new Fraction();
        f1.topN = 1;
        f1.btmN = 3;

        Fraction f2 = new Fraction();
        f2.topN = 4;
        f2.btmN = 16;

        Fraction f3 = new Fraction();
        f3.topN = 5;
        f3.btmN = 15;

        System.out.println(f1.toFloat());
        System.out.println(f2.toFloat());
        System.out.println(f3.toFloat());

        System.out.println("f1 is equal to f2 >> " + f1.myEquals(f2));
        System.out.println("f1 is equal to f3 >> " + f1.myEquals(f3));

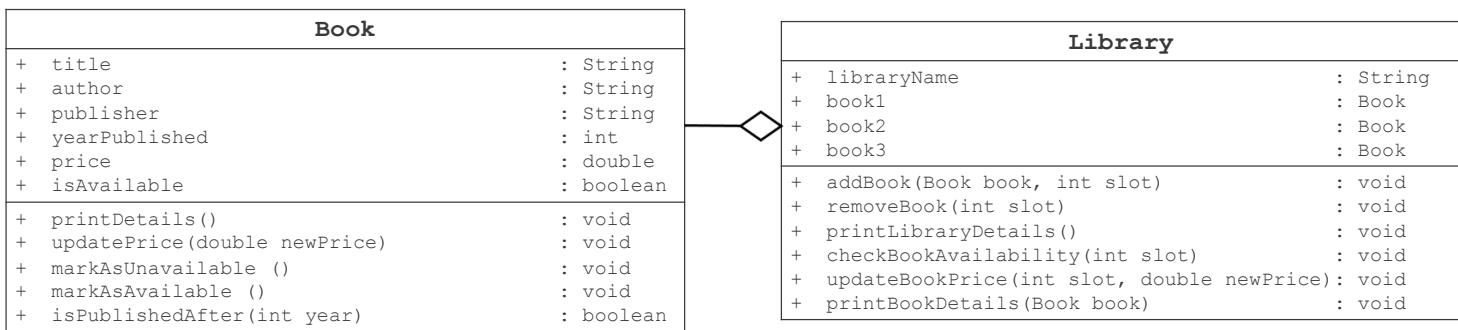
        System.out.println("Before : " + f2.toFraction());
        f2.LowestTermFrac();
        System.out.println("After : " + f2.toFraction());

    }
}
```

ตัวอย่างผลลัพธ์

```
0.3333333333333333
0.25
0.3333333333333333
f1 is equal to f2 >> false
f1 is equal to f3 >> true
Before : 4/16
After : 1/4
```

## 6. ให้รหัศึกษาสร้างคลาสตามไดอะแกรม Book และ Library ตามที่กำหนดให้ต่อไปนี้



โดยกำหนดให้คลาส Book มีรายละเอียดดังต่อไปนี้

- ให้รหัศึกษาสร้างแอ็พทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- ให้สร้างเมธอด printDetails() เพื่อแสดงข้อมูลต่อไปนี้

```
Title: [ค่าแอ็พทริบิวท์title]
Author: [ค่าแอ็พทริบิวท์author]
Publisher: [ค่าแอ็พทริบิวท์publisher]
Year Published: [ค่าแอ็พทริบิวท์yearPublished]
Price: $ [ค่าแอ็พทริบิวท์price]
Available: [ค่าแอ็พทริบิวท์isAvailable]
```

อย่างไรก็ตาม การแสดงค่าแอ็พทริบิวท์ isAvailable จะแสดง No ก็ต่อเมื่อแอ็พทริบิวท์ isAvailable มีค่าเป็น false ขณะที่ จะแสดงเป็น Yes ก็ต่อเมื่อแอ็พทริบิวท์ isAvailable มีค่าเป็น true

- ให้สร้างเมธอด updatePrice(double newPrice) ทำหน้าที่นำค่าจากพารามิเตอร์ newPrice นำไปกำหนดให้แอ็พทริบิวท์ price
- ให้สร้างเมธอด markAsUnavailable () ทำหน้าที่เปลี่ยนค่าแอ็พทริบิวท์ isAvailable เป็นค่า false
- ให้สร้างเมธอด markAsAvailable () ทำหน้าที่เปลี่ยนค่าแอ็พทริบิวท์ isAvailable เป็นค่า true
- ให้สร้างเมธอด isPublishedAfter(int year) จะคืนค่า true ก็ต่อเมื่อค่าพารามิเตอร์ year มีค่าน้อยกว่าค่าแอ็พทริบิวท์ yearPublished ขณะที่ จะคืนค่า false ก็ต่อเมื่อค่าพารามิเตอร์ year มีค่ามากกว่าหรือเท่ากับค่าแอ็พทริบิวท์ yearPublished

โดยกำหนดให้คลาส Library มีรายละเอียดดังต่อไปนี้

- ให้รหัศึกษาสร้างแอ็พทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- ให้สร้างเมธอด addBook(Book book, int slot) เพื่อจัดเก็บหนังสือเข้าห้องสมุด โดยจะนำพารามิเตอร์ book ที่รับเข้ามาไปเก็บไว้ใน
  - แอ็พทริบิวท์ book1 ก็ต่อเมื่อค่าพารามิเตอร์ slot มีค่าเท่ากับ 1

- แออททริบิวท์ book2 ก็ต่อเมื่อค่าพารามิเตอร์ slot มีค่าเท่ากับ 2
- แออททริบิวท์ book3 ก็ต่อเมื่อค่าพารามิเตอร์ slot มีค่าเท่ากับ 3
- ให้สร้างเมธอด removeBook(int slot) เพื่อลบหนังสือออกจากห้องสมุด โดยกำหนดให้
  - แออททริบิวท์ book1 มีค่าเป็น null ก็ต่อเมื่อค่าพารามิเตอร์ slot มีค่าเท่ากับ 1
  - แออททริบิวท์ book2 มีค่าเป็น null ก็ต่อเมื่อค่าพารามิเตอร์ slot มีค่าเท่ากับ 2
  - แออททริบิวท์ book3 มีค่าเป็น null ก็ต่อเมื่อค่าพารามิเตอร์ slot มีค่าเท่ากับ 3
- ให้สร้างเมธอด printLibraryDetails() เพื่อแสดงข้อมูลตามต่อไปนี้

Library: [ค่าแออททริบิวท์ libraryName]

Title: [ค่าแออททริบิวท์ title ของแออททริบิวท์ book1]  
 Author: [ค่าแออททริบิวท์ author ของแออททริบิวท์ book1]  
 Publisher: [ค่าแออททริบิวท์ publisher ของแออททริบิวท์ book1]  
 Year Published: [ค่าแออททริบิวท์ yearPublished ของแออททริบิวท์ book1]  
 Price: \$ [ค่าแออททริบิวท์ price ของแออททริบิวท์ book1]  
 Available: [ค่าแออททริบิวท์ isAvailable ของแออททริบิวท์ book1]

...

Title: [ค่าแออททริบิวท์ title ของแออททริบิวท์ book3]  
 Author: [ค่าแออททริบิวท์ author ของแออททริบิวท์ book3]  
 Publisher: [ค่าแออททริบิวท์ publisher ของแออททริบิวท์ book3]  
 Year Published: [ค่าแออททริบิวท์ yearPublished ของแออททริบิวท์ book3]  
 Price: \$ [ค่าแออททริบิวท์ price ของแออททริบิวท์ book3]  
 Available: [ค่าแออททริบิวท์ isAvailable ของแออททริบิวท์ book3]

อย่างไรก็ตาม การแสดงค่าแออททริบิวท์ isAvailable จะแสดง No ก็ต่อเมื่อแออททริบิวท์ isAvailable มีค่าเป็น false ขณะที่ จะแสดงเป็น Yes ก็ต่อเมื่อแออททริบิวท์ isAvailable มีค่าเป็น true (ดูตัวอย่างประกอบ)

- ให้สร้างเมธอด checkBookAvailability(int slot) เพื่อตรวจสอบว่าหนังสือใน slot ดังกล่าว ว่างหรือไม่ โดยโปรแกรมจะแสดงข้อความข้างล่าง ก็ต่อเมื่อค่าแออททริบิวท์ Book ตัวที่ slot ไม่เท่ากับ null

[ค่าแออททริบิวท์ title ของแออททริบิวท์ book ตัวที่ slot] is available.

ถ้าค่าแออททริบิวท์ Book ตัวที่ slot เท่ากับ null จะแสดงข้อความดังต่อไปนี้

Book in slot [ค่าพารามิเตอร์ slot] is not available.

- ให้สร้างเมธอด updateBookPrice(int slot, double newPrice) เพื่อปรับเปลี่ยนราคานั้นสือ โดยโปรแกรมจะแสดงข้อความข้างล่าง ก็ต่อเมื่อค่าแออททริบิวท์ Book ตัวที่ slot ไม่เท่ากับ null

Updated price of [ค่าแออททริบิวท์ title ของแออททริบิวท์ book ตัวที่ slot] to \$ [ค่าพารามิเตอร์ newPrice].

ถ้าค่าของทรีบิวท์ Book ตัวที่ slot เท่ากับ null จะแสดงข้อความดังต่อไปนี้

No book in this slot.

- ให้สร้างเมธอด printBookDetails (Book book) เพื่อแสดงรายละเอียดของหนังสือ โดยโปรแกรมจะแสดงข้อความข้างล่าง ก็ต่อเมื่อค่าพารามิเตอร์ Book ไม่เท่ากับ null

Title: [ค่าของทรีบิวท์ title]

Author: [ค่าของทรีบิวท์ author]

Publisher: [ค่าของทรีบิวท์ publisher]

Year Published: [ค่าของทรีบิวท์ yearPublished]

Price: \$ [ค่าของทรีบิวท์ price]

Available: [ค่าของทรีบิวท์ isAvailable]

ถ้าค่าพารามิเตอร์ Book เท่ากับ null จะแสดงข้อความดังต่อไปนี้

No book in this slot.

โค๊ดสำหรับทดสอบความถูกต้องของคลาส Book

```
public class Main {
    public static void main(String[] args) {
        Book book1 = new Book();
        book1.title = "Java Programming";
        book1.price = 450;
        book1.publisher = "Head First";
        book1.yearPublished = 2006;
        book1.isAvailable = false;
        book1.author = "John Smith";

        book1.printDetails();
        System.out.println("-----");
        book1.updatePrice(250);
        book1.markAsAvailable();
        book1.printDetails();
        System.out.println(book1.isPublishedAfter(2000));
    }
}
```

ตัวอย่างผลลัพธ์

Title: Java Programming

Author: John Smith

Publisher: Head First

Year Published: 2006

Price: \$450.0

Available: No

-----

Title: Java Programming

Author: John Smith

Publisher: Head First

Year Published: 2006

Price: \$250.0

Available: Yes

true

### โค้ดสำหรับทดสอบความถูกต้องของคลาส Library

```
public class Main {
    public static void main(String[] args) {
        Book book1 = new Book();
        book1.title = "Java Programming";
        book1.price = 450;
        book1.publisher = "Head First";
        book1.yearPublished = 2006;
        book1.isAvailable = false;
        book1.author = "John Smith";

        Book book2 = new Book();
        book2.title = "Python Programming";
        book2.price = 225;
        book2.publisher = "KM";
        book2.yearPublished = 2020;
        book2.isAvailable = true;
        book2.author = "Elon Potter";

        Library lib = new Library();
        lib.libraryName = "IT Library";
        lib.addBook(book2, 1);
        lib.addBook(book1, 3);

        lib.updateBookPrice(2, 1000);
        lib.updateBookPrice(3, 320);

        lib.printLibraryDetails();
    }
}
```

### ตัวอย่างผลลัพธ์

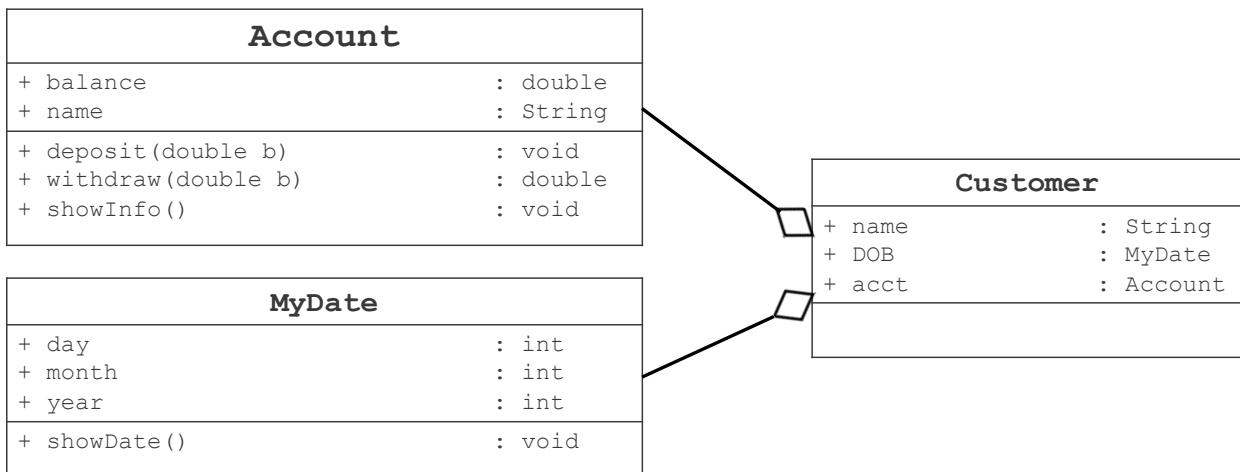
```
No book in this slot.
Updated price of Java Programming to $320.0.
Library: IT Library

Title: Python Programming
Author: Elon Potter
Publisher: KM
Year Published: 2020
Price: $225.0
Available: Yes

No book in this slot.

Title: Java Programming
Author: John Smith
Publisher: Head First
Year Published: 2006
Price: $320.0
Available: No
```

## 7. ให้นักศึกษาสร้างคลาสตามไดอะแกรมต่อไปนี้



โดยกำหนดให้คลาส Account มีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างแออททริบิวท์ตามที่กำหนดในคลาสไไดอะแกรม
- เมธอด **deposit(...)** ทำหน้าที่เพิ่มค่าแออททริบิวท์ **balance** ตามค่าในตัวแปร **b** ที่รับเข้ามา เมื่อตัวแปร **b** มีค่ามากกว่าหรือเท่ากับ 0 แต่ถ้าไม่ใช่จะแสดงข้อความต่อไปนี้

The balance variable must be greater than or equal to zero.

- เมธอด **withdraw(...)** ทำหน้าที่ลดค่าแออททริบิวท์ **balance** ตามค่าในตัวแปร **b** ที่รับเข้ามา จากนั้น จะดำเนินการคืนค่า (return) เงินที่ถอนออกมา เมื่อตัวแปร **b** มีค่ามากกว่าหรือเท่ากับ 0 แต่ถ้าไม่ใช่จะคืนค่าเป็น 0 พร้อมแสดงข้อความต่อไปนี้

The balance variable must be greater than or equal to zero.

นอกจากนี้ ค่าแออททริบิวท์ **balance** ต้องไม่ติดลบ แต่ถ้าไม่ใช่ค่าแออททริบิวท์ **balance** จะไม่ถูกลดพร้อมคืนค่าเป็น 0 และแสดงข้อความต่อไปนี้

Your account balance is insufficient.

- เมธอด **showInfo(...)** ทำหน้าที่แสดงข้อความต่อไปนี้

In [ค่าแออททริบิวท์ **name**] account, there is a balance equal to [ค่าแออททริบิวท์ **balance**]  
baht.

โดยกำหนดให้คลาส MyDate มีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างแออททริบิวท์ตามที่กำหนดในคลาสไไดอะแกรม
- เมธอด **showDate(...)** ทำหน้าที่แสดงข้อความต่อไปนี้

[ค่าแออททริบิวท์ **day**]/[ค่าแออททริบิวท์ **month**]/[ค่าแออททริบิวท์ **year**].

โดยกำหนดให้คลาส Customer มีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างแอ็พทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม

โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาสที่นักศึกษาได้พัฒนาขึ้นมา

```
public class Main {
    public static void main(String[] args) {
        Customer c = new Customer();
        c.name = "Taravichet";

        MyDate dob = new MyDate();
        dob.day = 8;
        dob.month = 11;
        dob.year = 2023;

        Account acct = new Account();
        acct.name = c.name;
        acct.balance = 500;

        c.DOB = dob;
        c.acct = acct;

        System.out.println("My name is " + c.name + ".");
        c.acct.showInfo();
        c.DOB.showDate();

        c.acct.deposit(500);
        c.acct.showInfo();
        c.acct.withdraw(3000);
        c.acct.showInfo();

    }
}
```

ตัวอย่างผลลัพธ์

```
My name is Taravichet.
In Taravichet account, there is a balance equal to 500.0 baht.
8/11/2023
In Taravichet account, there is a balance equal to 1000.0 baht.
Your account balance is insufficient.
In Taravichet account, there is a balance equal to 1000.0 baht.
```

ให้นักศึกษาอธิบายความหมายของคำสั่งต่อไปนี้

```
c.DOB.showDate();

c.acct.deposit(500);
c.acct.showInfo();
c.acct.withdraw(3000);
c.acct.showInfo();
```