

บทที่ 6 : การพัฒนาโปรแกรมภาษาจาวาเชิงอ็อบเจกต์

วัตถุประสงค์

- เพื่อให้เข้าใจหลักการเขียนโปรแกรมเชิงอ็อบเจกต์
- เพื่อให้เข้าใจความหมายของเมธอดแบบ overloaded และเมธอดแบบ overridden
- เพื่อให้เข้าใจความหมายของคลาสชนิด abstract และอินเตอร์เฟส (Interface)

แบบฝึกหัดที่ 1 โปรแกรมจำลองระบบธนาคาร

วัตถุประสงค์

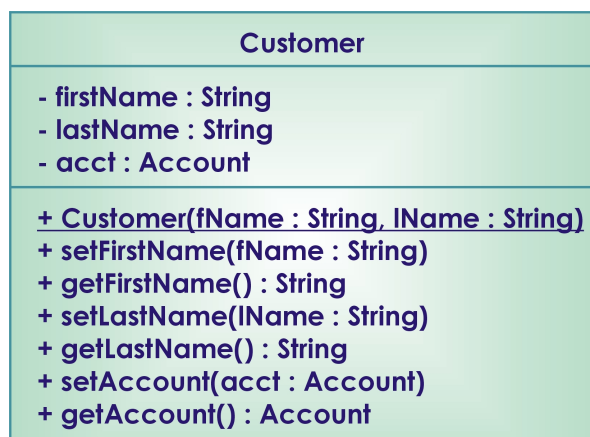
- 1) เพื่อให้เข้าใจหลักการการเขียนโปรแกรมเชิงอ็อบเจกต์
- 2) เพื่อให้เข้าใจการพัฒนาโปรแกรมจากไดอะแกรมของคลาสที่กำหนดมาให้

คำสั่ง

นำโปรแกรมเดิมจากแบบฝึกหัดเชิงปฏิบัติการที่ 3 และ 4 ของบทที่ 4 ซึ่งก็คือคลาส Account และคลาส CheckingAccount มา จากนั้นเพิ่มเติมคลาสที่ชื่อ Customer ที่มีคุณลักษณะคือ ชื่อ นามสกุลและบัญชีธนาคาร โดยมีเมธอดในการกำหนดค่า (setter) ของคุณลักษณะ และดึงค่า (getter) จากคุณลักษณะเหล่านี้

ขั้นตอนการปฏิบัติการ

- 1) จงเขียนคลาส Customer เพิ่มเติม ที่มีคลาสไดอะแกรมดังแสดงในรูปที่ 6.1



รูปที่ 6.1 ไดอะแกรมของคลาส Customer

โดยมีขั้นตอนดังนี้

- (a) กำหนดคุณลักษณะของคลาส Customer ที่ชื่อ firstName, lastName และ acct ให้มีชนิดข้อมูลแบบ String, String และ Account ตามลำดับ โดยมี access modifier เป็นแบบ private
 - (b) เขียน constructor ที่มี argument เพื่อรับข้อมูลที่เป็นชื่อของลูกค้า
 - (c) เขียนเมธอดแบบ setter เพื่อกำหนดค่าให้กับคุณลักษณะทั้งสาม
 - (d) เขียนเมธอดแบบ getter เพื่อเรียกดูค่าของคุณลักษณะทั้งสาม
- 2) จงเขียนคลาส Teller2 ที่มีเมธอด main() อยู่ภายในคลาสโดยมีคำสั่งต่างๆดังนี้

- (a) คำสั่งประกาศและสร้างอ็อบเจกต์ของคลาส `CheckingAccount` ที่ชื่อ `myAccount` โดยมีเงินเปิดบัญชี 4,000 บาทและมี credit 500 บาท โดยใช้คำสั่ง

```
Account myAccount;  
myAccount = new CheckingAccount(4000, 500);
```
- (b) คำสั่งประกาศและสร้างอ็อบเจกต์ของคลาส `Customer` ที่ชื่อ `cust` โดยมีชื่อเจ้าของบัญชีเป็น "Somchai Sommut" โดยใช้คำสั่ง

```
Customer cust;  
  
cust = new Customer("Somchai", "Sommut");
```
- (c) เรียกใช้เมธอด `setAccount()` เพื่อกำหนดคุณลักษณะที่ชื่อ `acct` ของอ็อบเจกต์ `cust` ให้มีค่าเป็น `myAccount`
- (d) เรียกใช้เมธอด `withdraw()` จากคำสั่ง `cust.getAccount()` เพื่อดถอนเงินในบัญชีของอ็อบเจกต์ที่ชื่อ `myAccount` จำนวน 4,300 บาท
- (e) เขียนคำสั่งที่เรียกใช้เมธอด `showBalance()` และ `showCredit()` จากคำสั่ง

```
cust.getAccount()
```

 เพื่อแสดงยอดเงินและ credit ในบัญชีของอ็อบเจกต์ที่ชื่อ `myAccount`
- (f) เรียกใช้เมธอด `deposit()` จากคำสั่ง `cust.getAccount()` เพื่อฝากเงินในบัญชีของอ็อบเจกต์ที่ชื่อ `myAccount` จำนวน 4,200 บาท
- (g) เขียนคำสั่งที่เรียกใช้เมธอด `showBalance()` และ `showCredit()` จากคำสั่ง

```
cust.getAccount()
```

 เพื่อแสดงยอดเงินและ credit ในบัญชีของอ็อบเจกต์ที่ชื่อ `myAccount`
- (h) เขียนคำสั่งเพื่อแสดงชื่อและนามสกุลของลูกค้าที่มีอ็อบเจกต์ที่ชื่อ `cust`
- (i) คอมไพล์และรันโปรแกรมจากคลาสที่ชื่อ `Teller2.java`

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 และเปรียบเทียบกับอ็อบเจกต์ชนิด `ComplexNumber` ตัวอื่น
3. จงเขียนคลาสที่ชื่อ `Employee` ซึ่งมีคุณลักษณะสามตัวคือ `name`, `position` และ `salary` โดยเขียนโปรแกรมที่ใช้หลักการของการห่อหุ้ม กำหนดเมธอดที่ชื่อ `calSalary()` ซึ่งรับข้อมูลคือจำนวนชั่วโมงการทำงานชนิด `int` และอัตราค่าจ้างต่อชั่วโมงชนิด `double` เข้ามาและส่งผลคูณของค่าทั้งสองกลับไปเป็นชนิด `double` จากนั้นให้กำหนดเมธอด `showDetails()` เพื่อแสดงรายละเอียดของคุณลักษณะทั้งสาม และกำหนดเมธอด `main()` ในการสร้างอ็อบเจกต์ของคลาส `Employee` และทดลองเรียกใช้เมธอดต่างๆ
4. จากคลาส `Employee` จงเขียนคลาสที่ชื่อ `Manager` ซึ่งสืบทอดมาจากคลาส `Employee` และมีคุณลักษณะที่ชื่อ `department` เพิ่มขึ้นมา และ `override` เมธอด `showDetails()` และกำหนดเมธอด `main()` ในการสร้างอ็อบเจกต์ชนิด `Manager`
5. จงเขียนโปรแกรมข้อ 4 ใหม่ โดยกำหนดให้ `Employee` เป็นอินเตอร์เฟซ และกำหนดคุณลักษณะทั้งหมดในคลาส `Manager`