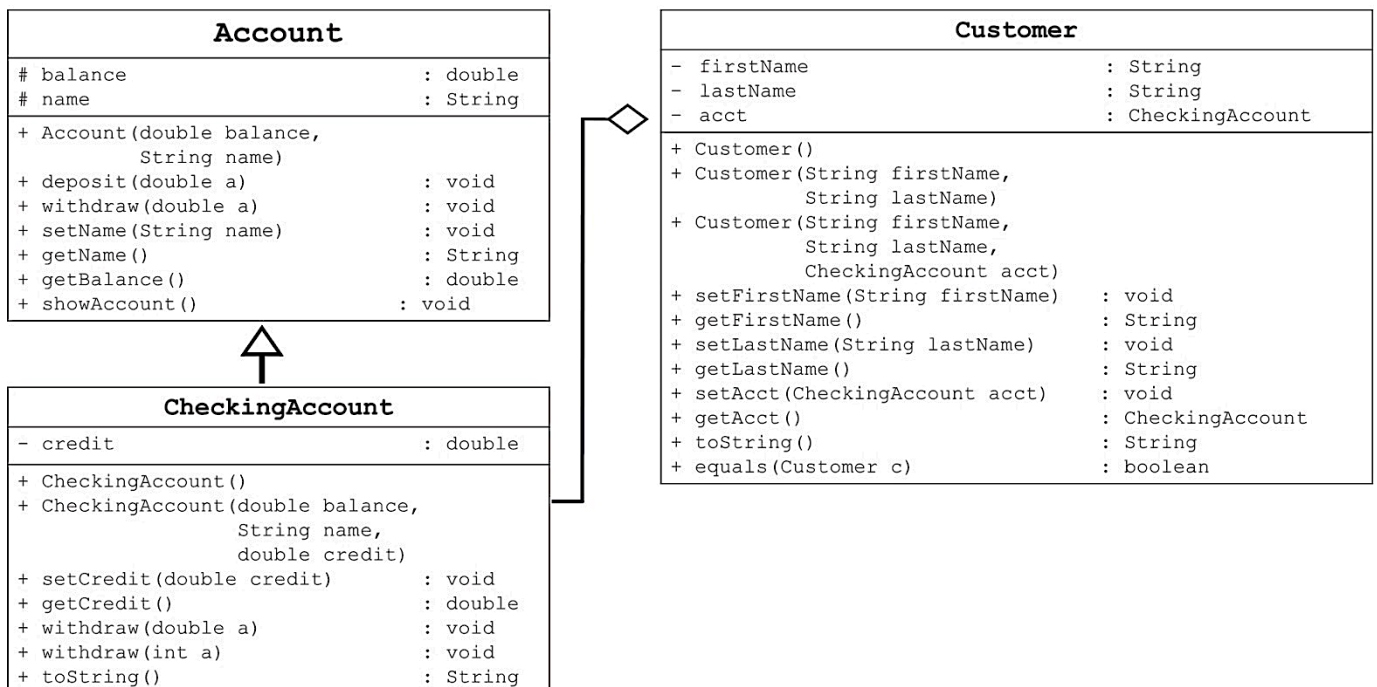


แบบฝึกปฏิบัติ ครั้งที่ 8

- เรื่อง อะเรย์และคอลเล็กชัน
- วัตถุประสงค์
1. เพื่อให้เข้าใจประโยชน์ของการใช้อะเรย์และคอลเล็กชัน
 2. เพื่อให้เข้าใจวิธีการประกาศและสร้างอะเรย์ของข้อมูลชนิดพื้นฐานและชนิดคลาส
 3. เพื่อให้เข้าใจคลาสที่อยู่ใน Collection API เช่น ArrayList และการนำไปใช้งาน

จากแบบฝึกปฏิบัติ ครั้งที่ 6 นักศึกษาได้สร้างคลาส Account, CheckingAccount และ Customer ตามคลาสไดอะแกรม ดังนี้



ซึ่งนักศึกษาต้องนำมาปรับปรุงในแบบฝึกหัดครั้งนี้ ตามรายละเอียด ดังนี้

แบบฝึกหัดที่ 1 การเขียนโปรแกรมธนาคาร

ขั้นตอนการปฏิบัติการ

- 1) คัดลอกไฟล์ Account.java และ CheckingAccount.java ที่พัฒนาขึ้นในบทที่ 5
- 2) เขียนคลาสที่ชื่อ Bank ซึ่งมีไดอะแกรมของคลาสดังแสดงในรูปที่ 9.1

Bank	
- acct[]	: Account
- numAcct	: int
+ Bank()	
+ addAccount(Account ac)	: void
+ getAccount(int index)	: Account
+ getNumAccount()	: int

รูปที่ 9.1 ไดอะแกรมของคลาส Bank

โดยมีขั้นตอนดังนี้

- กำหนดคุณลักษณะชื่อ acct ให้เป็นอะเรย์ของคลาส Account เพื่อใช้เก็บข้อมูลบัญชีทั้งหมดของธนาคาร
 - กำหนดคุณลักษณะชื่อ numAcct เป็นชนิดข้อมูลแบบ int เพื่อใช้เก็บจำนวนบัญชีของธนาคาร
 - เขียน default constructor ของคลาส Bank
 - ใช้คำสั่ง new ทำการสร้างอะเรย์ acct โดยกำหนดให้มีจำนวนสมาชิก 10 ตัว
 - เขียนเมธอดที่ชื่อ addAccount(Account ac)
 - ทำการเพิ่มบัญชี ac (อ็อบเจกต์ของคลาส Account ที่รับเข้ามา) เข้าไปในตัวแปรอะเรย์ acct
 - ทำการเพิ่มค่า numAcct ขึ้น 1 เนื่องจากตัวแปรอะเรย์ acct มีจำนวนอ็อบเจกต์บัญชีเพิ่มขึ้น
 - เขียนเมธอดที่ชื่อ getAccount(int index)
 - ทำการคืนค่าบัญชี (อ็อบเจกต์ของคลาส Account) ซึ่งเป็นสมาชิกของตัวแปรอะเรย์ acct ที่ตำแหน่ง index
 - เขียนเมธอด getNumAccount()
 - ทำการคืนค่าจำนวนบัญชี (numAcct)
- 3) ปรับปรุงคลาส Main เพื่อให้สามารถจัดการกับอ็อบเจกต์ของคลาส Bank ที่มีบัญชีหลายบัญชีได้โดยใช้คำสั่งดังนี้
 - กำหนดเมธอด main() ในคลาส Main
 - สร้างอ็อบเจกต์ myBank ของคลาส Bank
 - สร้างอ็อบเจกต์ที่ชื่อ acct1 ของคลาส Account โดยมีเงินเปิดบัญชี 5,000 บาท และมีชื่อบัญชี Sommai

- สร้างอ็อบเจกต์ที่ชื่อ acct2 ของคลาส Account โดยมีเงินเปิดบัญชี 3,000 บาท และมีชื่อบัญชี Somchai
- สร้างอ็อบเจกต์ที่ชื่อ acct3 ของคลาส Account โดยมีเงินเปิดบัญชี 900 บาท และมีชื่อบัญชี Somsri
- เรียกใช้เมธอด addAccount(Account ac) เพื่อใส่อ็อบเจกต์ของคลาส Account ทั้งหมด (acct1, acct2, acct3) ให้กับอ็อบเจกต์ myBank
- เรียกใช้เมธอดเพื่อทดลองฝากและถอนเงินในบัญชีแต่ละบัญชีและแสดงยอดเงินคงเหลือออกมา
- ทำการแสดงรายละเอียดบัญชีทั้งหมดที่มีใน myBank

ตัวอย่างโค้ดในเมธอด main() เป็นดังนี้ นศ.สามารถทดลองปรับปรุงโค้ดได้ตามความเหมาะสม

```
public static void main(String[] args) {
    Account a1 = new Account(5000, "Sommai");
    Account a2 = new Account(3000, "Somchai");
    Account a3 = new Account(900, "Somsri");

    Bank myBank = new Bank();
    myBank.addAccount(a1);
    myBank.addAccount(a2);
    myBank.addAccount(a3);

    System.out.println("Number of Account: " + myBank.getNumAccount());
    myBank.getAccount(0).showAccount();
    System.out.println("Deposit 500...");
    myBank.getAccount(0).deposit(500);
    myBank.getAccount(0).showAccount();
    System.out.println("Show all accounts");

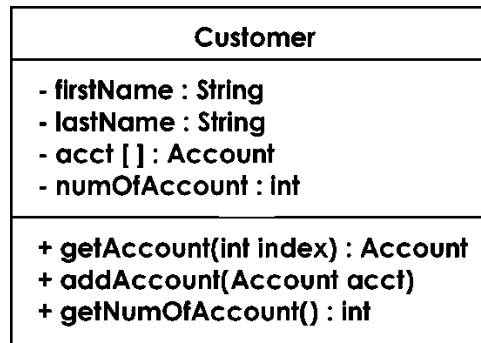
    System.out.println("=====");
    for(int i=0; i<myBank.getNumAccount(); i++){
        myBank.getAccount(i).showAccount();
    }
}
```

แบบฝึกหัดที่ 2 การใช้ตัวแปรของข้อมูลชนิดคลาส

คำสั่ง จากโปรแกรมจำลองระบบธนาคาร ให้ทดลองปรับปรุงคลาส Customer เพื่อกำหนดให้มีคุณลักษณะที่สามารถเก็บบัญชีธนาคารได้หลายบัญชี โดยกำหนดเป็น ตัวแปรชนิดอะเรย์

ขั้นตอนการปฏิบัติการ

- 1) ปรับปรุงคลาส Customer เพื่อให้มีคลาสไดอะแกรมดังแสดงในรูปที่ 9.2



รูปที่ 9.2 ไดอะแกรมของคลาส Customer

โดยมีขั้นตอนดังนี้

- เปลี่ยนคุณลักษณะ acct ให้เป็นอะเรย์ของข้อมูลชนิด Account เพื่อให้ลูกค้ามีบัญชีหลายบัญชีได้
- เพิ่มคุณลักษณะ numOfAccount ที่เป็นข้อมูลชนิด int เพื่อเก็บจำนวนบัญชีของลูกค้า
- เขียนเมธอด addAccount(Account acct)
 - ทำการเพิ่มบัญชี acct (อ็อบเจกต์ของคลาส Account ที่รับเข้ามา) เข้าไปในตัวแปรอะเรย์ acct
 - ทำการเพิ่มค่า numOfAccount ขึ้น 1 เนื่องจากตัวแปรอะเรย์ acct มีจำนวนอ็อบเจกต์บัญชีเพิ่มขึ้น
- ลบ constructor Customer(String firstName, String lastName, CheckingAccount acct) ออก
- ปรับปรุง constructor Customer (String firstName, String lastName) และ Customer()
 - เพิ่มคำสั่งสร้างอะเรย์ของคลาส Account ที่ชื่อ acct ให้มีจำนวนสมาชิก 5 ตัว ดังนี้


```
acct = new Account[5];
```
- ปรับปรุงเมธอด toString() ให้แสดง ชื่อ-นามสกุล ของลูกค้า และจำนวนบัญชีที่ลูกค้ามี
- เขียนเมธอด getAccount(int index)
 - ทำการคืนค่าบัญชี (อ็อบเจกต์ของคลาส Account) ซึ่งเป็นสมาชิกของตัวแปรอะเรย์ acct ที่ตำแหน่ง index
- เขียนเมธอด getNumOfAccount()
 - ทำการคืนค่าของคุณลักษณะ numOfAccount ซึ่งเป็นจำนวนของบัญชีทั้งหมดที่ลูกค้ามีอยู่

2) ปรับปรุงคลาส Main เพื่อให้สามารถจัดการกับอ็อบเจกต์ของคลาส Customer ที่มีบัญชีหลายบัญชีได้โดยใช้คำสั่งดังนี้

- กำหนดเมธอด main() ในคลาส Main
- สร้างอ็อบเจกต์ cust ของคลาส Customer
- สร้างอ็อบเจกต์ที่ชื่อ acct1 ของคลาส Account โดยมีเงินเปิดบัญชี 5,000 บาท (กำหนดชื่อบัญชีเอง)
- สร้างอ็อบเจกต์ที่ชื่อ acct2 ของคลาส Account โดยมีเงินเปิดบัญชี 3,000 บาท (กำหนดชื่อบัญชีเอง)
- เรียกใช้เมธอด addAccount(Account acct) เพื่อใส่อ็อบเจกต์ของคลาส Account ทั้งสอง (acct1 และ acct2) ให้กับอ็อบเจกต์ cust
- เรียกใช้เมธอดเพื่อทดลองฝากและถอนเงินในบัญชีแต่ละบัญชีและแสดงยอดเงินคงเหลือออกมา
- แสดงรายละเอียดของ cust ว่ามีชื่อ-นามสกุลอะไร และมีบัญชีอยู่กับบัญชี
- ทำการแสดงรายละเอียดบัญชีทั้งหมดที่ cust มีอยู่

ตัวอย่างโค้ดในเมธอด main() เป็นดังนี้ นศ.สามารถทดลองปรับปรุงโค้ดได้ตามความเหมาะสม

```
public static void main(String[] args) {
    Customer cust = new Customer("Somsri", "Boonjing");
    Account acct1 = new Account(5000, "Somsri01");
    Account acct2 = new Account(3000, "Somsri02");
    cust.addAccount(acct1);
    cust.addAccount(acct2);
    // ทดลองฝากเงิน ถอนเงินในบัญชีต่างๆ
    cust.getAccount(0).withdraw(3000);
    cust.getAccount(1).deposit(3000);
    // แสดงข้อมูลของลูกค้า เช่น Somsri Boonjing has 2 accounts.
    System.out.println(cust);
    // ทดลองสร้างบัญชีและเพิ่มบัญชีอื่นๆ ให้กับลูกค้า มากกว่า 5 บัญชี
    for (int i = 0; i < cust.getNumOfAccount(); i++) {
        cust.getAccount(i).showAccount();
    }
}
```

แบบฝึกหัดที่ 3 การใช้ Collection API

คำสั่ง จากโปรแกรมจำลองระบบธนาคาร ให้ทดลองปรับปรุงคลาส Customer เพื่อกำหนดให้มีคุณลักษณะที่สามารถเก็บบัญชีธนาคารได้หลายบัญชี โดยกำหนดให้เป็นข้อมูลแบบ ArrayList

ขั้นตอนการปฏิบัติการ

- 1) ปรับปรุงคลาส Customer เพื่อกำหนดคุณลักษณะ acct ให้เป็นชนิดข้อมูลแบบ ArrayList โดยมีขั้นตอนดังนี้
 - เปลี่ยนคุณลักษณะ acct ให้เป็นอ็อบเจกต์ของคลาส ArrayList โดยใช้คำสั่ง


```
ArrayList acct;
```
 - เปลี่ยนคำสั่งใน constructor Customer (String firstName, String lastName) และ Customer() เพื่อสร้างอ็อบเจกต์ acct เป็น


```
acct = new ArrayList()
```
 - ปรับปรุงเมธอด addAccount(Account acct)
 - ในส่วนของการเพิ่มบัญชี acct (อ็อบเจกต์ของคลาส Account ที่รับเข้ามา) เข้าไปในตัวแปร ArrayList acct โดยเรียกใช้เมธอด add() ในคลาส ArrayList
 - ปรับปรุงเมธอด getAccount(int index)
 - ให้เรียกใช้เมธอด get() ในคลาส ArrayList เพื่อคืนค่าสมาชิกที่ตำแหน่ง index ของ acct คืนมา
 - ปรับปรุงเมธอด getNumOfAccount()
 - ให้เรียกใช้เมธอด size() ในคลาส ArrayList เพื่อคืนค่าจำนวนบัญชีทั้งหมดที่ลูกค้ามีอยู่
- 2) คอมไพล์และรันโปรแกรม Main.java ใหม่