การทดลองที่ 3

Chapter 11 Inheritance and Polymorphism

① **Programming Exercise 11.8 p.446** (New Account class)

An Account class was specified in Programming Exercise 9.7. Design a new Account class as follows:

- Add a new data field name of the String type to store the name of the customer.
- Add a new constructor that constructs an account with the specified name, id, and balance.
- Add a new data field named transactions whose type is ArrayList that stores the transaction for the accounts. Each transaction is an instance of the Transaction class. The Transaction class is defined as shown in Figure 11.6.

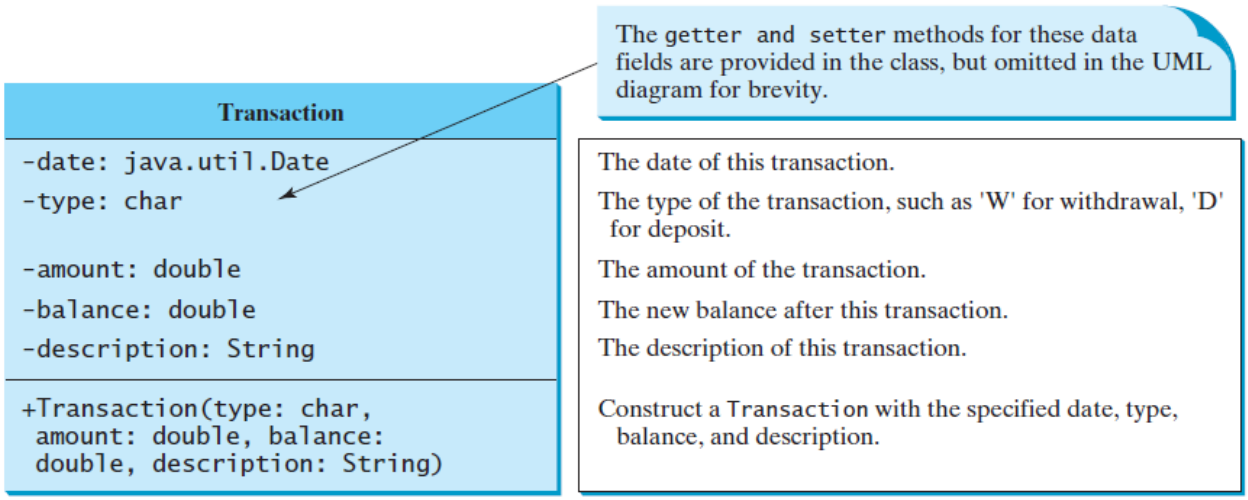

FIGURE 11.6 The Transaction class describes a transaction for a bank account.

- Modify the withdraw and deposit methods to add a transaction to the transactions array list.
- All other properties and methods are the same as in Programming Exercise 9.7.

Write a test program that creates an Account with annual interest rate 1.5%, balance 1000, id 1122, and name George. Deposit $30, $40, and $50 to the account and withdraw $5, $4, and $2 from the account. Print an account summary that shows account holder name, interest rate, balance, and all transactions.

---

② **Programming Exercise 11.3 p.445** (Subclasses of Account)

In Programming Exercise 9.7, the Account class was defined to model a bank account. An account has the properties account id, balance, annualInterestRate, and dateCreated, and methods to deposit and withdraw funds.

Create 3 subclasses for saving, checking and privileged accounts. A privileged account has an overdraft limit, but savings and checking account cannot be overdrawn. Checking account does not compute any interest for the balance.

Draw the UML diagram for the classes and then implement them. Write a test program that creates objects of Account, SavingsAccount, CheckingAccount and PriviledgedAccount and invokes their toString() methods.

③ **Programming Exercise 11.2 p.445**

(The Person, Student, Employee, Faculty, and Staff classes)

Design a class named Person and its two subclasses named Student and Employee. Make Faculty and Staff subclasses of Employee.

- A person has a name, address, phone number, and email address.
- A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant.
- An employee has an office, salary, and date hired. Use the MyDate class defined in Programming Exercise 10.14 to create an object for date hired.
- A faculty member has office hours and a rank.
- A staff member has a title.
- Override the toString method in each class to display the class name and the person's name.

Draw the UML diagram for the classes and implement them. Write a test program that creates a Person, Student, Employee, Faculty, and Staff, and invokes their toString() methods.
