Labs

Lab 6.1: The Account Class

What is the purpose?

In this lab, you will write a banking application that tracks deposits and withdrawals and accrues interest using the following methodology:
Design a class named Account that contains:

- An int data field named id for the account (default 0)
- A double data field named balance for the account (default 0)
- A double data field named annualInterestRate that stores the current interest rate (default 0)
- A Date data field named dateCreated that stores the date when the account was created
- A no-arg constructor that creates a default account
- The accessor and mutator methods for id, balance, and annualInterestRate
- The accessor method for dateCreated
- A method named getMonthlyInterestRate() that returns the monthly interest rate
- A method named withDraw that withdraws a specified amount from the account
- A method named deposit that deposits a specified amount from the account

Write a test program that creates an Account object with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $10,000, use the deposit method to deposit $3,000, and print the balance, the monthly interest, and the date when the account was created.

What are the steps?

- Task 1:

  Procedure

  1. Create an abstract class called Account.
  2. Create the following methods:
     - public int id;
     - public double balance;
     - public double annualInterestRate;
     - public java.util.Date dateCreated;
     - public Account()
     - public Account(int id, double balance, double annualInterestRate)
     - public int getId()
     - public double getBalance()
     - public double getAnnualInterestRate()
     - public void setId(int id)
     - public void setBalance(double balance)
     - public void setAnnualInterestRate(double annualInterestRate)
     - public double getMonthlyInterest()
     - public java.util.Date getDateCreated()
     - public void withdraw(double amount)

- public void deposit(double amount)

3. Create a test program to implement the Account class and its properties. See Figure 6-1-1.

```
public static void main (String[] args) {
  Account account = new Account(1122, 20000, 4.5);

   account.withdraw(10000);
   account.deposit(3000);
   System.out.println("Balance is " + account.getBalance());
   System.out.println("Monthly interest is " +
     account.getMonthlyInterest());
   System.out.println("This account was created at " +
     account.getDateCreated());
 }
```

Figure 6-1-1

4. Express the result to the console.
5. Compile the java file using the javac command.
6. Execute the java class using the java command.
7. Save a screen shot of the output similar to Figure 6-1-2 and submit it to your

```
Balance is 13000.0
Monthly interest is 48.75
This account was created at Tue Dec 12 12:03:36 PST 2006
```

Figure 6-1-2

Did it work?

Were you able to—

- Implement the Account class?
- Display an account’s balance amount by calling the getBalance() method?
- Display an account’s monthly interest rate by calling the getMonthlyInterest() method?
- Display an account’s created time by calling the getDateCreated() method?

Lab 6.2: Subclasses of Account Class

What is the purpose?

In Lab 6.1, you created the Account class to model a bank account. An account has the properties account number, balance, annual interest rate, date created, and methods to deposit and withdraw. In this exercise, you will create two subclasses for checking and saving accounts. Limit the withdrawal amount to be less than or equal to $10,000 for checking account per each transaction, and display an alert message.

Write a test program called SavingsDemo that creates Account and SavingsAccount objects with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $10,000, use the deposit method to deposit $3,000, and print the account type, the balance, the monthly interest, alert message, and the date when the account was created.

Write another test program called CheckingDemo that creates Account and CheckingAccount objects with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $10,000, use the deposit method to deposit $3,000, and print the account type, the balance, the monthly interest, alert message, and the date when the account was created.

What are the steps?

- Task 1:

  Procedure

  1. Create a Java class called Checking that extends Account class.
  2. Create the following methods for both Checking classes:
     - public String accountType()
     - public int getId()
     - public double getBalance()
     - public double getAnnualInterestRate()
     - public void setId(int id)
     - public void setBalance(double balance)
     - public void setAnnualInterestRate(double annualInterestRate)
     - public double getMonthlyInterest()
     - public java.util.Date getDateCreated()
     - public void withdraw(double amount)
     - public void deposit(double amount)
     - public String getAlert(double amount)
  3. Create a Java class called Savings that extends Account class.
  4. Create the following methods for both Savings classes:
     - public String accountType()
     - public int getId()
     - public double getBalance()
     - public double getAnnualInterestRate()
     - public void setId(int id)
     - public void setBalance(double balance)
     - public void setAnnualInterestRate(double annualInterestRate)
     - public double getMonthlyInterest()
     - public java.util.Date getDateCreated()
     - public void withdraw(double amount)
     - public void deposit(double amount)
     - public String getAlert(double amount)

5. Write a demo program to invoke the CheckingAccount and its methods as pictured in Figure 6-2-1.

```
public class CheckingDemo {
  public static void main (String[] args) {
    Checking check = new Checking(1122, 20000, 4.5);

    check.withdraw(10000);
    check.deposit(3000);
    System.out.println("Account type is " + check.accountType());
    System.out.println("Balance is " + check.getBalance());
    System.out.println("Monthly interest is " +
      check.getMonthlyInterest());
    System.out.println(check.getAlert(10000));
    System.out.println("This account was created at " +
      check.getDateCreated());
  }
}
```

Figure 6-2-1

6. Write a demo program to invoke the SavingsAccount and its methods pictured in Figure 6-2-2.

```
public class SavingsDemo {
  public static void main (String[] args) {
    Savings save = new Savings(1122, 20000, 4.5);

    save.withdraw(10000);
    save.deposit(3000);
    System.out.println("Account type is " + save.accountType());
    System.out.println("Balance is " + save.getBalance());
    System.out.println("Monthly interest is " +
      save.getMonthlyInterest());
    System.out.println(save.getAlert(10000));
    System.out.println("This account was created at " +
      save.getDateCreated());
  }
}
```

Figure 6-2-2

7. Express the result to the console.
8. Compile the java file using the javac command.
9. Execute the java class using the java command.
10. Save screenshots of the output similar to Figures 6-2-3 and 6-2-4 and submit them to your instructor.

```
Account type is Checking
Balance is 23000.0
Monthly interest is 86.25
** Withdrawn amount = 10000.0 is over the $10,000 limit **
This account was created at Sun Jul 26 01:43:30 PDT 2009
```

Figure 6-2-3: Sample Output of Checking Account

```
Account type is Savings
Balance is 13000.0
Monthly interest is 48.75
No alert message.
This account was created at Sun Jul 26 01:45:11 PDT 2009
```

Figure 6-2-4: Sample Output of Savings Account

Did it work?

Were you able to—

- Implement the CheckingDemo class?
- Display an checking account's balance amount by calling the getBalance() method?
- Display an checking account's monthly interest rate by calling the getMonthlyInterest() method?
- Display an checking account's over-withdrawal message by calling the getAlert() method?
- Display an checking account's created time by calling the getDateCreated() method?
- Implement the SavingsDemo class?
- Display an savings account's balance amount by calling the getBalance() method?
- Display an savings account's monthly interest rate by calling the getMonthlyInterest() method?
- Display an savings account's over-withdrawal message by calling the getAlert() method?
- Display an savings account's created time by calling the getDateCreated() method?