

Lab Assignment 07

2022 – 2023 Spring, SENG102

Q1. File *Account.java* contains a definition for a simple bank account class with methods to withdraw, deposit, get the balance and account number, and return a String representation.

Save this class to your directory and modify it as follows:

1. Overload the constructor as follows:

- `public Account (double initBal, String owner, long number)` - initializes the balance, owner, and account number as specified
- `public Account (double initBal, String owner)` - initializes the balance and owner as specified; randomly generates the account number.
- `public Account (String owner)` - initializes the owner as specified; sets the initial balance to 0 and randomly generates the account number.

2. Overload the *withdraw* method with one that also takes a fee and deducts that fee from the account.

Complete in the below test program "*TestAccount.java*" so that it will

- a. Prompt for and read the account holder's name and create an account
- b. Prompt for and read the initial balance for this account holder and creates a new account
- c. Prompt for and read the account number for the account holder and create a new account with this account number
- d. Deposit 100\$ into account and prints the new balance
- e. Withdraw 25\$ with 2\$ fee and prints the new balance

```
import java.util.Scanner;
public class TestAccount
{
    public static void main(String[] args)
    {
        String name;
        double balance;
        long acctNum;
        Account acct;
        Scanner scan = new Scanner(System.in);

        System.out.println("Enter account holder's first name");

        // a.Read the account holder's name and create acct

        System.out.println("Account for " + name + ":");
        System.out.println(acct);

        System.out.println("\nEnter initial balance");

        //b.Read the initial balance for the account holder and create a new account

        System.out.println("Account for " + name + ":");
        System.out.println(acct);
```

```

        //c.Prompt for and read the account number for the account holder and create a new
        account with this account number

        System.out.println("\nEnter account number");
        System.out.println("Account for " + name + ":");
        System.out.println(acct);

        System.out.print("\nDepositing 100 into account, balance is now ");

        //d.Deposit 100$ into account and prints the new balance

        System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");
        //e. Withdraw 25$ with 2$ fee and print the new balance

    }

}

```

Sample Output:

Enter account holder's first name

John

Account for John:

Name: John

Acct #: 330692253

Balance: 0.0

Enter initial balance

500

Account for John:

Name: John

Acct #: 129015726

Balance: 500.0

Enter account number

25431

Account for John:

Name: John

Acct #: 25431

Balance: 500.0

Depositing 100 into account, balance is now 600.0

Withdrawing \$25 with \$2 fee, balance is now 573.0

Q2. Use your “*Account.java*” class that you implemented in Q1.

1. Add a static method *Account consolidate(Account acct1, Account acct2)* to your *Account* class that creates a new account whose balance is the sum of the balances in *acct1* and *acct2* and account number is sum of the account numbers.

Two important rules of consolidation:

- Only accounts with the same name can be consolidated. The new account gets the same name but a new account number.
- Two accounts with the same account number cannot be consolidated.

Check these conditions before creating the new account. If either condition fails, do not create the new account; print a useful message and return null.

2. Write a test program “*TestAccounts2.java*” that prompts for and reads in three names and creates an account with an initial balance of \$ 100 for each. The account numbers should be 1000, 1100 and 1500 respectively. Print the three accounts and try to consolidate the second and third into a new account. If consolidation is made, print the consolidated account.

Sample Output:

```
Enter name for first account: Jess
First account
Name: Jess
Acct #: 1000
Balance: 100.0

Enter name for second account: Rory
Second account
Name: Rory
Acct #: 1100
Balance: 100.0

Enter name for third account: Elisa
Third account
Name: Elisa
Acct #: 1500
Balance: 100.0
Trying to consolidate second and third accounts.
Sorry, accounts with different names cannot be consolidated.
```

```
Enter name for first account: Rocky
```

First account
Name: Rocky
Acct #: 1000
Balance: 100.0

Enter name for second account: Jessie
Second account
Name: Jessie
Acct #: 1100
Balance: 100.0

Enter name for third account: Jessie
Third account
Name: Jessie
Acct #: 1500
Balance: 100.0
Trying to consolidate second and third accounts.
Result account is
Name: Jessie
Acct #: 2600
Balance: 200.0

***** Upload Account.java and TestAccounts2.java files on webonline.**