

```

1  //*****
2  //  Account.java          Author: Lewis/Loftus/Cocking
3  //
4  //  Represents a bank account with basic services such as deposit
5  //  and withdraw.
6  //*****
7
8  import java.text.NumberFormat;
9
10 public class Account
11 {
12     private NumberFormat fmt = NumberFormat.getCurrencyInstance();
13
14     private final double RATE = 0.035; // interest rate of 3.5%
15
16     private int acctNumber;
17     private double balance;
18     private String name;
19
20     //-----
21     //  Sets up the account by defining its owner, account number,
22     //  and initial balance.
23     //-----
24     public Account (String owner, int account, double initial)
25     {
26         name = owner;
27         acctNumber = account;
28         balance = initial;
29     }
30
31     //-----
32     //  Validates the transaction, then deposits the specified amount
33     //  into the account. Returns the new balance.
34     //-----
35     public double deposit (double amount)
36     {
37         if (amount < 0) // deposit value is negative
38         {
39             System.out.println ();
40             System.out.println ("Error: Deposit amount is invalid.");
41             System.out.println (acctNumber + " " + fmt.format(amount));
42         }
43         else
44             balance = balance + amount;
45         return balance;
46     }
47
48     //-----
49     //  Validates the transaction, then withdraws the specified amount
50     //  from the account. Returns the new balance.
51     //-----
52     public double withdraw (double amount, double fee)
53     {
54         amount += fee;
55     }

```

VISIBILITY

TYPES: PRIMITIVE

CLASSES

RETURN TYPE

METHODS

VARIABLES: LOCAL

PARAMETER

INSTANCE

CLASS

```

56     if (amount < 0) // withdraw value is negative
57     {
58         System.out.println ();
59         System.out.println ("Error: Withdraw amount is invalid.");
60         System.out.println ("Account: " + acctNumber);
61         System.out.println ("Requested: " + fmt.format(amount));
62     }
63     else
64         if (amount > balance) // withdraw value exceeds balance
65         {
66             System.out.println ();
67             System.out.println ("Error: Insufficient funds.");
68             System.out.println ("Account: " + acctNumber);
69             System.out.println ("Requested: " + fmt.format(amount));
70             System.out.println ("Available: " + fmt.format(balance));
71         }
72     else
73         balance = balance - amount;
74
75     return balance;
76 }
77
78 //-----
79 // Adds interest to the account and returns the new balance.
80 //-----
81 public double addInterest ()
82 {
83     balance += (balance * RATE);
84     return balance;
85 }
86
87 //-----
88 // Returns the current balance of the account.
89 //-----
90 public double getBalance ()
91 {
92     return balance;
93 }
94
95 //-----
96 // Returns the account number.
97 //-----
98 public int getAccountNumber ()
99 {
100     return acctNumber;
101 }
102
103 //-----
104 // Returns a one-line description of the account as a string.
105 //-----
106 public String toString ()
107 {
108     return (acctNumber + "\t" + name + "\t" + fmt.format(balance));
109 }
110 }

```