

SE 317, Lab 6

Name: Aina Qistina Binti Azman

Net ID: 457 464 051

Screenshots of the program, ATMandUtility.java

Checking Account Focus

1. Choosing Bank Account: Checking Account & Login

The screenshot shows a Java IDE interface with the file `ATMandUtility.java` open. The code defines a class `ATMandUtility` with static fields for bank and utility accounts, and protected static fields for checking and saving files. Below the code, a terminal window displays the execution of the program. It asks for account type (1/2) and logs in successfully. It then asks for a bank account and prompts for username and password. Finally, it asks for an account type again.

```
ATMandUtility.java X
1 package lab6;
2
3import java.io.File;
4
5 public class ATMandUtility {
6
7     protected static String bank_username;
8     protected static String bank_password;
9
10    protected static String utility_username;
11    protected static String utility_password;
12
13    protected static String checkingFile = "CheckingAccount.txt";
14    protected static String savingFile = "SavingAccount.txt";
15
16    protected static CheckingAccount checking = null;
17    protected static SavingAccount saving = null;
18
19
20 }
```

Console X
ATMandUtility [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (25 Jun 2024, 11:55:23 pm)
1. Logging on to Bank Accounts
2. Logging on to Utility Company Account
Account type (1/2):
1

Bank Account
Enter your username:
user123
Enter your password:
summer2024
Successfully login into your Bank Account!
1. Checking Account
2. Saving Account
Account type (1/2):

2. Initial Checking Balance = \$1000

Input '1' for Deposit = \$300

Current Checking Balance = \$1300

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a folder named "ioveyuboss" containing sub-folders "Lab 2", "Lab 3", "Lab 4", "Lab 5", and "Lab 6".
- Code Editor:** Displays the `ATMandUtility.java` file with the following code snippet:


```

14
15     protected static String checkingFile = "CheckingAccount.txt";
16
17     protected static String savingfile = "SavingAccount.txt";
18
19     protected static CheckingAccount checking = null;
20     protected static SavingAccount saving = null;
21
22     public static void main(String[] args) {
23
24         int accounttype;
25
26         Scanner scanner = new Scanner(System.in);
27
28         System.out.println("1. Logging on to Bank Accounts");
29         System.out.println("2. Logging on to Utility Company Accounts");
      
```
- Console Output:** Shows the program's execution and user interaction. The user has chosen activity 5 (Checking Account Balance) twice, resulting in a balance of \$1300.0.

```

14
15     protected static String checkingFile = "CheckingAccount.txt";
16
17     protected static String savingfile = "SavingAccount.txt";
18
19     protected static CheckingAccount checking = null;
20     protected static SavingAccount saving = null;
21
22     public static void main(String[] args) {
23
24         int accounttype;
25
26         Scanner scanner = new Scanner(System.in);
27
28         System.out.println("1. Logging on to Bank Accounts");
29         System.out.println("2. Logging on to Utility Company Accounts");
30
31         System.out.print("Choose Activity (1/2/3/4/5/6/7): ");
32         accounttype = scanner.nextInt();
33
34         if (accounttype == 1) {
35             System.out.print("Enter Checking Account File Name: ");
36             checkingFile = scanner.nextLine();
37
38             System.out.print("Enter Saving Account File Name: ");
39             savingfile = scanner.nextLine();
40
41             CheckingAccount checking = new CheckingAccount();
42             SavingAccount saving = new SavingAccount();
43
44             checking.readData(checkingFile);
45             saving.readData(savingfile);
46
47             System.out.print("Enter Account Number: ");
48             String accountNumber = scanner.nextLine();
49
50             CheckingAccount checkingAccount = checking.getAccount(accountNumber);
51             SavingAccount savingAccount = saving.getAccount(accountNumber);
52
53             if (checkingAccount != null && savingAccount != null) {
54                 System.out.println("Welcome to ATM and Utility Company!");
55
56                 System.out.print("1. Deposit Money");
57                 System.out.print("2. Withdraw Money");
58                 System.out.print("3. Transfer to Saving Account");
59                 System.out.print("4. Pay Bills");
60                 System.out.print("5. Checking Account Balance");
61                 System.out.print("6. Log on to Utility Company");
62                 System.out.print("7. Quit");
63
64                 System.out.print("Choose Activity (1/2/3/4/5/6/7): ");
65                 accounttype = scanner.nextInt();
66
67                 switch (accounttype) {
68                     case 1:
69                         System.out.print("Enter Amount to Deposit: ");
70                         double amountToDeposit = scanner.nextDouble();
71                         checkingAccount.deposit(amountToDeposit);
72                         System.out.println("Balance: $" + checkingAccount.getBalance());
73                         break;
74
75                     case 2:
76                         System.out.print("Enter Amount to Withdraw: ");
77                         double amountToWithdraw = scanner.nextDouble();
78                         checkingAccount.withdraw(amountToWithdraw);
79                         System.out.println("Balance: $" + checkingAccount.getBalance());
80                         break;
81
82                     case 3:
83                         System.out.print("Enter Amount to Transfer: ");
84                         double amountToTransfer = scanner.nextDouble();
85                         checkingAccount.transfer(savingAccount, amountToTransfer);
86                         System.out.println("Balance: $" + checkingAccount.getBalance());
87                         System.out.println("New Balance: $" + savingAccount.getBalance());
88                         break;
89
90                     case 4:
91                         System.out.print("Enter Bill Amount: ");
92                         double billAmount = scanner.nextDouble();
93                         checkingAccount.payBills(billAmount);
94                         System.out.println("Bill Paid: $" + billAmount);
95                         System.out.println("Remaining Balance: $" + checkingAccount.getBalance());
96                         break;
97
98                     case 5:
99                         System.out.println("Current Checking Account Balance: $" + checkingAccount.getBalance());
100                        break;
101
102                     case 6:
103                         System.out.println("Logging on to Utility Company Accounts");
104                         break;
105
106                     case 7:
107                         System.out.println("Thank you for using our services!");
108                         break;
109
110                     default:
111                         System.out.println("Invalid Choice!");
112                         break;
113                 }
114             } else {
115                 System.out.println("Account Not Found!");
116             }
117         } else if (accounttype == 2) {
118             System.out.print("Enter Utility Company Account File Name: ");
119             savingfile = scanner.nextLine();
120
121             SavingAccount saving = new SavingAccount();
122
123             saving.readData(savingfile);
124
125             System.out.print("Enter Account Number: ");
126             String accountNumber = scanner.nextLine();
127
128             SavingAccount savingAccount = saving.getAccount(accountNumber);
129
130             if (savingAccount != null) {
131                 System.out.println("Welcome to Utility Company!");
132
133                 System.out.print("1. Deposit Money");
134                 System.out.print("2. Withdraw Money");
135                 System.out.print("3. Transfer to Saving Account");
136                 System.out.print("4. Pay Bills");
137                 System.out.print("5. Checking Account Balance");
138                 System.out.print("6. Log on to Utility Company");
139                 System.out.print("7. Quit");
140
141                 System.out.print("Choose Activity (1/2/3/4/5/6/7): ");
142                 accounttype = scanner.nextInt();
143
144                 switch (accounttype) {
145                     case 1:
146                         System.out.print("Enter Amount to Deposit: ");
147                         double amountToDeposit = scanner.nextDouble();
148                         savingAccount.deposit(amountToDeposit);
149                         System.out.println("Balance: $" + savingAccount.getBalance());
150                         break;
151
152                     case 2:
153                         System.out.print("Enter Amount to Withdraw: ");
154                         double amountToWithdraw = scanner.nextDouble();
155                         savingAccount.withdraw(amountToWithdraw);
156                         System.out.println("Balance: $" + savingAccount.getBalance());
157                         break;
158
159                     case 3:
160                         System.out.print("Enter Amount to Transfer: ");
161                         double amountToTransfer = scanner.nextDouble();
162                         savingAccount.transfer(checkingAccount, amountToTransfer);
163                         System.out.println("Balance: $" + savingAccount.getBalance());
164                         System.out.println("New Balance: $" + checkingAccount.getBalance());
165                         break;
166
167                     case 4:
168                         System.out.print("Enter Bill Amount: ");
169                         double billAmount = scanner.nextDouble();
170                         savingAccount.payBills(billAmount);
171                         System.out.println("Bill Paid: $" + billAmount);
172                         System.out.println("Remaining Balance: $" + savingAccount.getBalance());
173                         break;
174
175                     case 5:
176                         System.out.println("Current Saving Account Balance: $" + savingAccount.getBalance());
177                         break;
178
179                     case 6:
180                         System.out.println("Logging on to Utility Company Accounts");
181                         break;
182
183                     case 7:
184                         System.out.println("Thank you for using our services!");
185                         break;
186
187                     default:
188                         System.out.println("Invalid Choice!");
189                         break;
190                 }
191             } else {
192                 System.out.println("Account Not Found!");
193             }
194         }
195     }
196 }
```

3. Input '2' for Withdraw = \$400

Current Checking Balance = \$900

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a folder named "ioveyuboss" containing sub-folders "Lab 2", "Lab 3", "Lab 4", "Lab 5", and "Lab 6".
- Code Editor:** Displays the `ATMandUtility.java` file with the same code as the previous screenshot, but the console output shows a withdrawal of \$400 from the checking account, resulting in a balance of \$900.0.
- Console Output:** Shows the program's execution and user interaction. The user has chosen activity 5 (Checking Account Balance) twice, resulting in a balance of \$900.0.

```

14
15     protected static String checkingFile = "CheckingAccount.txt";
16     protected static String savingfile = "SavingAccount.txt";
17
18     protected static CheckingAccount checking = null;
19     protected static SavingAccount saving = null;
20
21     public static void main(String[] args) {
22
23         int accounttype;
24
25         Scanner scanner = new Scanner(System.in);
26
27         System.out.println("1. Logging on to Bank Accounts");
28         System.out.println("2. Logging on to Utility Company Accounts");
29
30         System.out.print("Choose Activity (1/2/3/4/5/6/7): ");
31         accounttype = scanner.nextInt();
32
33         if (accounttype == 1) {
34             System.out.print("Enter Checking Account File Name: ");
35             checkingFile = scanner.nextLine();
36
37             System.out.print("Enter Saving Account File Name: ");
38             savingfile = scanner.nextLine();
39
40             CheckingAccount checking = new CheckingAccount();
41             SavingAccount saving = new SavingAccount();
42
43             checking.readData(checkingFile);
44             saving.readData(savingfile);
45
46             System.out.print("Enter Account Number: ");
47             String accountNumber = scanner.nextLine();
48
49             CheckingAccount checkingAccount = checking.getAccount(accountNumber);
50             SavingAccount savingAccount = saving.getAccount(accountNumber);
51
52             if (checkingAccount != null && savingAccount != null) {
53                 System.out.println("Welcome to ATM and Utility Company!");
54
55                 System.out.print("1. Deposit Money");
56                 System.out.print("2. Withdraw Money");
57                 System.out.print("3. Transfer to Saving Account");
58                 System.out.print("4. Pay Bills");
59                 System.out.print("5. Checking Account Balance");
60                 System.out.print("6. Log on to Utility Company");
61                 System.out.print("7. Quit");
62
63                 System.out.print("Choose Activity (1/2/3/4/5/6/7): ");
64                 accounttype = scanner.nextInt();
65
66                 switch (accounttype) {
67                     case 1:
68                         System.out.print("Enter Amount to Deposit: ");
69                         double amountToDeposit = scanner.nextDouble();
70                         checkingAccount.deposit(amountToDeposit);
71                         System.out.println("Balance: $" + checkingAccount.getBalance());
72                         break;
73
74                     case 2:
75                         System.out.print("Enter Amount to Withdraw: ");
76                         double amountToWithdraw = scanner.nextDouble();
77                         checkingAccount.withdraw(amountToWithdraw);
78                         System.out.println("Balance: $" + checkingAccount.getBalance());
79                         break;
80
81                     case 3:
82                         System.out.print("Enter Amount to Transfer: ");
83                         double amountToTransfer = scanner.nextDouble();
84                         checkingAccount.transfer(savingAccount, amountToTransfer);
85                         System.out.println("Balance: $" + checkingAccount.getBalance());
86                         System.out.println("New Balance: $" + savingAccount.getBalance());
87                         break;
88
89                     case 4:
90                         System.out.print("Enter Bill Amount: ");
91                         double billAmount = scanner.nextDouble();
92                         checkingAccount.payBills(billAmount);
93                         System.out.println("Bill Paid: $" + billAmount);
94                         System.out.println("Remaining Balance: $" + checkingAccount.getBalance());
95                         break;
96
97                     case 5:
98                         System.out.println("Current Checking Account Balance: $" + checkingAccount.getBalance());
99                         break;
100
101                     case 6:
102                         System.out.println("Logging on to Utility Company Accounts");
103                         break;
104
105                     case 7:
106                         System.out.println("Thank you for using our services!");
107                         break;
108
109                     default:
110                         System.out.println("Invalid Choice!");
111                         break;
112                 }
113             } else {
114                 System.out.println("Account Not Found!");
115             }
116         } else if (accounttype == 2) {
117             System.out.print("Enter Utility Company Account File Name: ");
118             savingfile = scanner.nextLine();
119
120             SavingAccount saving = new SavingAccount();
121
122             saving.readData(savingfile);
123
124             System.out.print("Enter Account Number: ");
125             String accountNumber = scanner.nextLine();
126
127             SavingAccount savingAccount = saving.getAccount(accountNumber);
128
129             if (savingAccount != null) {
130                 System.out.println("Welcome to Utility Company!");
131
132                 System.out.print("1. Deposit Money");
133                 System.out.print("2. Withdraw Money");
134                 System.out.print("3. Transfer to Saving Account");
135                 System.out.print("4. Pay Bills");
136                 System.out.print("5. Checking Account Balance");
137                 System.out.print("6. Log on to Utility Company");
138                 System.out.print("7. Quit");
139
140                 System.out.print("Choose Activity (1/2/3/4/5/6/7): ");
141                 accounttype = scanner.nextInt();
142
143                 switch (accounttype) {
144                     case 1:
145                         System.out.print("Enter Amount to Deposit: ");
146                         double amountToDeposit = scanner.nextDouble();
147                         savingAccount.deposit(amountToDeposit);
148                         System.out.println("Balance: $" + savingAccount.getBalance());
149                         break;
150
151                     case 2:
152                         System.out.print("Enter Amount to Withdraw: ");
153                         double amountToWithdraw = scanner.nextDouble();
154                         savingAccount.withdraw(amountToWithdraw);
155                         System.out.println("Balance: $" + savingAccount.getBalance());
156                         break;
157
158                     case 3:
159                         System.out.print("Enter Amount to Transfer: ");
160                         double amountToTransfer = scanner.nextDouble();
161                         savingAccount.transfer(checkingAccount, amountToTransfer);
162                         System.out.println("Balance: $" + savingAccount.getBalance());
163                         System.out.println("New Balance: $" + checkingAccount.getBalance());
164                         break;
165
166                     case 4:
167                         System.out.print("Enter Bill Amount: ");
168                         double billAmount = scanner.nextDouble();
169                         savingAccount.payBills(billAmount);
170                         System.out.println("Bill Paid: $" + billAmount);
171                         System.out.println("Remaining Balance: $" + savingAccount.getBalance());
172                         break;
173
174                     case 5:
175                         System.out.println("Current Saving Account Balance: $" + savingAccount.getBalance());
176                         break;
177
178                     case 6:
179                         System.out.println("Logging on to Utility Company Accounts");
180                         break;
181
182                     case 7:
183                         System.out.println("Thank you for using our services!");
184                         break;
185
186                     default:
187                         System.out.println("Invalid Choice!");
188                         break;
189                 }
190             } else {
191                 System.out.println("Account Not Found!");
192             }
193         }
194     }
195 }
```

4. Input '3' for transferring money to Saving Account = \$50

Current Checking Balance = \$850

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like Lab 1 through Lab 6.
- Code Editor:** Displays the `ATMandUtility.java` file containing Java code for a bank account management system.
- Terminal Window:** Shows the program's output. It starts with a welcome message, asks for account type, and lists activities. The user chooses activity 7 (Quit), and the program exits.

```

SE317 - Lab 6/src/lab6/ATMandUtility.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X Problems
ATMandUtility.java
14  protected static String checkingfile = "CheckingAccount.txt";
15  protected static String savingfile = "SavingAccount.txt";
16
17  protected static CheckingAccount checking = null;
18  protected static SavingAccount saving = null;
19
20  public static void main(String[] args) {
21
22      int accountType;
23
24      Scanner scanner = new Scanner(System.in);
25
26      System.out.println("1. Logging on to Bank Accounts");
27      System.out.println("2. Logging on to Utility Company Accounts");
28
29  Amount to transfer to Saving Account:
30  -----
31  1. Deposit Money
32  2. Withdraw Money
33  3. Transfer to Saving Account
34  4. Pay Bills
35  5. Checking Account Balance
36  6. Log on to Utility Company
37  7. Quit
38  Choose Activity (1/2/3/4/5/6/7):
39
40  Balance: $850.0
41  -----
42  1. Deposit Money
43  2. Withdraw Money
44  3. Transfer to Saving Account
45  4. Pay Bills
46  5. Checking Account Balance
47  6. Log on to Utility Company
48  7. Quit
49  Choose Activity (1/2/3/4/5/6/7):
50
Choose Activity (1/2/3/4/5/6/7):

```

5. Input '7' to Quit Program

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like Lab 1 through Lab 6.
- Code Editor:** Displays the `ATMandUtility.java` file containing Java code for a bank account management system.
- Terminal Window:** Shows the program's output. It starts with a welcome message, asks for account type, and lists activities. The user chooses activity 7 (Quit), and the program exits with a message: "Exiting the program. Thank you for using our service!"

```

SE317 - Lab 6/src/lab6/ATMandUtility.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X Problems
ATMandUtility.java
14  protected static String checkingfile = "CheckingAccount.txt";
15  protected static String savingfile = "SavingAccount.txt";
16
17  protected static CheckingAccount checking = null;
18  protected static SavingAccount saving = null;
19
20  public static void main(String[] args) {
21
22      int accountType;
23
24      Scanner scanner = new Scanner(System.in);
25
26      System.out.println("1. Logging on to Bank Accounts");
27      System.out.println("2. Logging on to Utility Company Accounts");
28
29  Amount to transfer to Saving Account:
30  -----
31  1. Deposit Money
32  2. Withdraw Money
33  3. Transfer to Saving Account
34  4. Pay Bills
35  5. Checking Account Balance
36  6. Log on to Utility Company
37  7. Quit
38  Choose Activity (1/2/3/4/5/6/7):
39
40  Balance: $850.0
41  -----
42  1. Deposit Money
43  2. Withdraw Money
44  3. Transfer to Saving Account
45  4. Pay Bills
46  5. Checking Account Balance
47  6. Log on to Utility Company
48  7. Quit
49  Choose Activity (1/2/3/4/5/6/7):
50
51  -----
52  1. Deposit Money
53  2. Withdraw Money
54  3. Transfer to Saving Account
55  4. Pay Bills
56  5. Checking Account Balance
57  6. Log on to Utility Company
58  7. Quit
59  Choose Activity (1/2/3/4/5/6/7):
60
61  Exiting the program. Thank you for using our service!

```

Saving Account Focus

1. Choosing Bank Account: Saving Account & Login

SE317 - Lab 6/src/lab6/ATMandUtility.java - Eclipse IDE

```

14  protected static String checkingfile = "CheckingAccount.txt";
15  protected static String savingfile = "SavingAccount.txt";
16
17  protected static CheckingAccount checking = null;
18  protected static SavingAccount saving = null;
19
20  public static void main(String[] args) {
21
22      int accountType;
23
24      Scanner scanner = new Scanner(System.in);
25
26      System.out.println("1. Logging on to Bank Accounts");
27      System.out.println("2. Logging on to Utility Company Account");
28
29
30      Bank Account
31      Enter your username:
32      se317user
33      Enter your password:
34      summer2024
35      Successfully login into your Bank Account!
36      1. Checking Account
37      2. Saving Account
38      Account type (1/2):
39      2
40      -----
41      1. Deposit Money
42      2. Transfer to Checking Account
43      3. Saving Account Balance
44      4. Quit
45      Choose Activity (1/2/3/4):
46

```

2. Initial Saving Balance = \$1000

Due to transfer from Checking Account previously

Current Saving Balance = \$1050

SE317 - Lab 6/src/lab6/ATMandUtility.java - Eclipse IDE

```

14  protected static String checkingfile = "CheckingAccount.txt";
15  protected static String savingfile = "SavingAccount.txt";
16
17  protected static CheckingAccount checking = null;
18  protected static SavingAccount saving = null;
19
20  public static void main(String[] args) {
21
22      int accountType;
23
24      Scanner scanner = new Scanner(System.in);
25
26      System.out.println("1. Logging on to Bank Accounts");
27      System.out.println("2. Logging on to Utility Company Account");
28
29
30      se317user
31      Enter your password:
32      summer2024
33      Successfully login into your Bank Account!
34      1. Checking Account
35      2. Saving Account
36      Account type (1/2):
37      2
38      -----
39      1. Deposit Money
40      2. Transfer to Checking Account
41      3. Saving Account Balance
42      4. Quit
43      Choose Activity (1/2/3/4):
44      3
45      Balance: $1050.0
46      -----
47      1. Deposit Money
48      2. Transfer to Checking Account
49      3. Saving Account Balance
50      4. Quit
51      Choose Activity (1/2/3/4):

```

3. Input '1' for deposit = \$60

Current Saving Balance = \$1110

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows projects Lab 1 through Lab 6.
- Code Editor:** Displays the `ATMandUtility.java` file content. The code initializes static variables for checking and saving files, and defines a main method that prints a menu and uses a scanner to handle user input.
- Terminal (Console):** Shows the program's execution. It prints two logging messages, then a menu with options 1 through 4. When option 1 is selected, it asks for an amount to deposit (\$60) and updates the balance to \$110.0.

```

SE317 - Lab 6/src/lab6/ATMandUtility.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X ATManUtility.java X
14
15     protected static String checkingfile = "CheckingAccount.txt";
16     protected static String savingfile = "SavingAccount.txt";
17
18     protected static CheckingAccount checking = null;
19     protected static SavingAccount saving = null;
20
21 public static void main(String[] args) {
22
23     int accountType;
24
25     Scanner scanner = new Scanner(System.in);
26
27     System.out.println("1. Logging on to Bank Accounts");
28     System.out.println("2. Logging on to Utility Company Account");
29
30
31     int choice = scanner.nextInt();
32
33     if (choice == 1) {
34         CheckingAccount checking = new CheckingAccount();
35         checking.readData();
36
37         int activity;
38
39         while (true) {
40             System.out.println("1. Deposit Money");
41             System.out.println("2. Transfer to Checking Account");
42             System.out.println("3. Saving Account Balance");
43             System.out.println("4. Quit");
44
45             activity = scanner.nextInt();
46
47             switch (activity) {
48                 case 1:
49                     double amount = scanner.nextDouble();
50                     checking.deposit(amount);
51                     break;
52
53                 case 2:
54                     double transferAmount = scanner.nextDouble();
55                     checking.transfer(transferAmount);
56                     break;
57
58                 case 3:
59                     System.out.println("Balance: " + checking.getBalance());
60                     break;
61
62                 case 4:
63                     System.out.println("Goodbye!");
64                     return;
65             }
66         }
67     } else if (choice == 2) {
68         SavingAccount saving = new SavingAccount();
69         saving.readData();
70
71         int activity;
72
73         while (true) {
74             System.out.println("1. Deposit Money");
75             System.out.println("2. Transfer to Checking Account");
76             System.out.println("3. Saving Account Balance");
77             System.out.println("4. Quit");
78
79             activity = scanner.nextInt();
80
81             switch (activity) {
82                 case 1:
83                     double amount = scanner.nextDouble();
84                     saving.deposit(amount);
85                     break;
86
87                 case 2:
88                     double transferAmount = scanner.nextDouble();
89                     saving.transfer(transferAmount);
90                     break;
91
92                 case 3:
93                     System.out.println("Balance: " + saving.getBalance());
94                     break;
95
96                 case 4:
97                     System.out.println("Goodbye!");
98                     return;
99             }
100        }
101    }
102 }

```

4. Input '2' for transferring money to Checking Account = \$67

Current Saving Balance = \$1043

And input '4' to Quit Program

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows projects Lab 1 through Lab 6.
- Code Editor:** Displays the `ATMandUtility.java` file content. The code is identical to the previous screenshot.
- Terminal (Console):** Shows the program's execution. It prints two logging messages, then a menu with options 1 through 4. When option 1 is selected, it asks for an amount to transfer to the checking account (\$67) and updates the saving account balance to \$1043.0.

```

SE317 - Lab 6/src/lab6/ATMandUtility.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X ATManUtility.java X
14
15     protected static String checkingfile = "CheckingAccount.txt";
16     protected static String savingfile = "SavingAccount.txt";
17
18     protected static CheckingAccount checking = null;
19     protected static SavingAccount saving = null;
20
21 public static void main(String[] args) {
22
23     int accountType;
24
25     Scanner scanner = new Scanner(System.in);
26
27     System.out.println("1. Logging on to Bank Accounts");
28     System.out.println("2. Logging on to Utility Company Account");
29
30
31     int choice = scanner.nextInt();
32
33     if (choice == 1) {
34         CheckingAccount checking = new CheckingAccount();
35         checking.readData();
36
37         int activity;
38
39         while (true) {
40             System.out.println("1. Deposit Money");
41             System.out.println("2. Transfer to Checking Account");
42             System.out.println("3. Saving Account Balance");
43             System.out.println("4. Quit");
44
45             activity = scanner.nextInt();
46
47             switch (activity) {
48                 case 1:
49                     double amount = scanner.nextDouble();
50                     checking.deposit(amount);
51                     break;
52
53                 case 2:
54                     double transferAmount = scanner.nextDouble();
55                     checking.transfer(transferAmount);
56                     break;
57
58                 case 3:
59                     System.out.println("Balance: " + checking.getBalance());
60                     break;
61
62                 case 4:
63                     System.out.println("Goodbye!");
64                     return;
65             }
66         }
67     } else if (choice == 2) {
68         SavingAccount saving = new SavingAccount();
69         saving.readData();
70
71         int activity;
72
73         while (true) {
74             System.out.println("1. Deposit Money");
75             System.out.println("2. Transfer to Checking Account");
76             System.out.println("3. Saving Account Balance");
77             System.out.println("4. Quit");
78
79             activity = scanner.nextInt();
80
81             switch (activity) {
82                 case 1:
83                     double amount = scanner.nextDouble();
84                     saving.deposit(amount);
85                     break;
86
87                 case 2:
88                     double transferAmount = scanner.nextDouble();
89                     saving.transfer(transferAmount);
90                     break;
91
92                 case 3:
93                     System.out.println("Balance: " + saving.getBalance());
94                     break;
95
96                 case 4:
97                     System.out.println("Goodbye!");
98                     return;
99             }
100        }
101    }
102 }

```

Utility Account Focus

1. Choosing Utility Account & Login

The screenshot shows the Eclipse IDE interface with the project 'SE317 - Lab 6/src/lab6' open. The 'ATMandUtility.java' file is selected in the Project Explorer. The code defines static variables for file paths ('checkingFile' and 'savingFile') and account objects ('checking' and 'saving'), and a main method that prints logging messages to the console. The 'Console' tab shows the output of the program execution:

```
ATMandUtility [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (26 Jun 2024, 12:16:14 am)
1. Logging on to Bank Accounts
2. Logging on to Utility Company Account
Account type (1/2):
2

Utility Company Account
Enter your username:
SE317User
Enter your password:
summer2024
Successfully login into your Utility Account!
-----
1. Check bill payment history
2. Make payment
To QUIT, Press any number
```

2. Input '1' to check payment history

The screenshot shows the Eclipse IDE interface with the project 'SE317 - Lab 6/src/lab6' open. The 'ATMandUtility.java' file is selected in the Project Explorer. The code defines static variables for file paths ('checkingFile' and 'savingFile') and account objects ('checking' and 'saving'), and a main method that prints logging messages to the console. The 'Console' tab shows the output of the program execution after selecting option 1:

```
ATMandUtility [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (26 Jun 2024, 12:16:14 am)
-----
1. Check bill payment history
2. Make payment
To QUIT, Press any number
1
|-----
Bill Payment History
January Bill, $24.00
Payment of $24.00
February Bill, $28.00
Payment of $28.00
March Bill, $29.00
Payment of $29.00
April Bill, $33.00
Are you planning to pay your bills?
Enter 1 for YES
Enter any number to EXIT
```

3. Input '1' to make payment

Log in into Checking Account

The screenshot shows the Eclipse IDE interface with the file `ATMandUtility.java` open. The code implements a banking application with bill payment and utility company account features. The execution output in the Console tab shows the application's interaction with the user:

```
March Bill, $29.00
Payment of $29.00

April Bill, $33.00

Are you planning to pay your bills?
Enter 1 for YES
Enter any number to EXIT
1

Bank Account
Enter your username:
se31User
Enter your password:
summer2024
Successfully login into your Bank Account!
1. Checking Account
2. Saving Account
Account type (1/2):
1

-----
1. Deposit Money
2. Withdraw Money
3. Transfer to Saving Account
4. Pay Bills
5. Checking Account Balance
6. Log on to Utility Company
7. Quit
Choose Activity (1/2/3/4/5/6/7):
```

4. Input '4' to Pay Bills = \$33

Login into Utility Account

Check Payment History and there will be a new statement in the file

"Payment of \$33"

The screenshot shows the Eclipse IDE interface with the file `ATMandUtility.java` open. The code implements a banking application with bill payment and utility company account features. The execution output in the Console tab shows the application's interaction with the user:

```
7. QUIT
Choose Activity (1/2/3/4/5/6/7):
6
Transferring to Utility Company Account...

Utility Company Account
Enter your username:
se31User
Enter your password:
summer2024
Successfully login into your Utility Account!
-----
1. Check bill payment history
2. Make payment
To QUIT, Press any number
1

-----
Bill Payment History

January Bill, $24.00
Payment of $24.00

February Bill, $28.00
Payment of $28.00

March Bill, $29.00
Payment of $29.00

April Bill, $33.00
Payment of $33.00

Are you planning to pay your bills?
Enter 1 for YES
Enter any number to EXIT
```

Screenshots of Tests Cases

1. ATMandUtiliyTestLogin.java

Test Cases of Login into Bank Account and Utility Account.

```
SE317 - Lab 6/test/lab6/ATMandUtiliyTestLogin.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
JUnit Project Explorer JUnit
Finished after 0.075 seconds
Runs: 4/4 Errors: 0 Failures: 0
lab6.ATMandUtiliyTestLogin [Runner: JUnit]
  testBankAccountWrongCredentials (0.009 s)
  testBankAccountLoginSuccessful (0.008 s)
  testUtilityAccountLoginSuccessful (0.005 s)
  testUtilityAccountWrongCredentials (0.001)

Failure Trace
```

```
1 package lab6;
2
3 import org.junit.After;
4
5 public class ATMandUtiliyTestLogin {
6
7     private final ByteArrayOutputStream outContent = new ByteArrayOutputStream();
8
9     @Before
10    public void setup() {
11        System.setOut(new PrintStream(outContent));
12    }
13
14    @After
15    public void tearDown() {
16        System.setOut(null);
17        System.setIn(System.in);
18    }
19
20    @Test
21    public void testBankAccountLoginSuccessful() {
22
23        String username = "se317user";
24        String password = "summer2024";
25
26        String input = username + "\n" + password + "\n";
27
28        System.setIn(new ByteArrayInputStream(input.getBytes()));
29
30        try {
31
32            ATMandUtiliy.BankAccountLogin();
33            String output = outContent.toString();
34
35            assertTrue(output.contains("Successfully login into your Bank Account!"));
36        }
37    }
38
39}
```

2. ATMandUtilityTestCheckingAccount.java

Test Cases of methods related to Checking Account.

```
SE317 - Lab 6/test/lab6/ATMandUtilityTestCheckingAccount.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
JUnit Project Explorer JUnit
Finished after 0.251 seconds
Runs: 12/12 Errors: 0 Failures: 0
lab6.ATMandUtilityTestCheckingAccount [Runner: JUnit]
  testBankAccountCheckingTransferInsufficient (0.000 s)
  testBankAccountCheckingBalance (0.000 s)
  testBankAccountCheckingWithdraw (0.011 s)
  testBankAccountCheckingWithdrawExceeds (0.001 s)
  testBankAccountCheckingDepositExceeds (0.010 s)
  testBankAccountCheckingPayBillsInsufficient (0.021 s)
  testBankAccountCheckingDeposit (0.021 s)
  testBankAccountCheckingOverdraft (0.004 s)
  testBankAccountCheckingTransfer (0.025 s)
  testBankAccountCheckingPayBills (0.023 s)
  testBankAccountCheckingWithdrawInsufficient (0.001 s)

Failure Trace
```

```
1 package lab6;
2
3 import static org.junit.Assert.*;
4
5 public class ATMandUtilityTestCheckingAccount {
6
7     private final ByteArrayOutputStream outContent = new ByteArrayOutputStream();
8
9     private final String bankAccountType = "1";
10    private final String quit = "#";
11
12    private final String testingCheckingFile = "testingCheckingAccount.txt";
13    private final String testingSavingFile = "testingSavingAccount.txt";
14    private final String testingBill = "testingBill.txt";
15
16    @Before
17    public void setup() {
18
19        initializeTestingFile();
20        initializeBill();
21
22        ATMandUtility.checkingFile = testingCheckingFile;
23        ATMandUtility.savingFile = testingSavingFile;
24        ATMandUtility.bill = testingBill;
25
26        System.setOut(new PrintStream(outContent));
27    }
28
29    private void initializeTestingFile() {
30        try {
31            PrintWriter initializeChecking = new PrintWriter(testingCheckingFile);
32            initializeChecking.println("Checking Balance = 1000.0");
33            initializeChecking.close();
34
35            PrintWriter initializeSaving = new PrintWriter(testingSavingFile);
36
```

3. ATMandUtilityTestSavingAccount.java

Test Cases of methods related to Saving Account.

SE317 - Lab 6/test/lab6/ATMandUtilityTestSavingAccount.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
JUnit Project Explorer JUnit
Finished after 0.134 seconds
Runs: 7/7 Errors: 0 Failures: 0
Tab: ATMandUtilityTestSavingAccount [Runner]
  testBankAccountSavingTransferInsufficient (0.005 s)
  testBankAccountSavingBalance (0.005 s)
  testBankAccountExitProgram (0.008 s)
  testBankAccountSavingDepositExceedsMax (0.012 s)
  testBankAccountSavingTransferExceedsMax (0.021 s)
  testBankAccountSavingTransfer (0.021 s)
  testBankAccountSavingDeposit (0.018 s)
  testBankAccountSavingDeposit (0.018 s)

Failure Trace
  Console × Problems
  terminated: ATMandUtilityTestSavingAccount [JUnit] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (26 Jun 2024, 12:33:08 am - 12:33:09 am)

```

4. ATMandUtilityTestDataStorage.java

Test Cases of Data Storages.

SE317 - Lab 6/test/lab6/ATMandUtilityTestDataStorage.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
JUnit Project Explorer JUnit
Finished after 0.125 seconds
Runs: 7/7 Errors: 0 Failures: 0
Tab: ATMandUtilityTestDataStorage [Runner]
  testNormalMultipleElements (0.005 s)
  testNullSingleElement (0.009 s)
  testEmptyElements (0.009 s)
  testIncompatibleTypes (0.009 s)
  testNormalSingleElement (0.008 s)
  testNullStorage (0.004 s)
  testNullElementWithMultipleElements (0.004 s)

Failure Trace
  Console × Problems
  terminated: ATMandUtilityTestDataStorage [JUnit] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (26 Jun 2024, 12:33:57 am - 12:33:59 am)

```