

# Banking App Program using Collections with Import & Export Screenshots

## 1. Add Accounts

The screenshot shows the BankingApp.java file with the main method and a while loop for the menu. The console output shows the program running and the user adding two accounts: Simi and Shiny.

```
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
5 public class BankingApp {
6     private static Scanner in;
7     private static AccountService accountService;
8
9     public static void main(String[] args) {
10         in = new Scanner(System.in);
11         accountService = new AccountServiceHashMapImpl();
12         System.out.println("Welcome to Banking Application!");
13         while (true) {
14             System.out.println("\n");
15             System.out.println("1. Add Accounts");
16             System.out.println("2. View All Accounts");
17             System.out.println("3. View Account");
18             System.out.println("4. Update Account");
19             System.out.println("5. Delete Account");
20             System.out.println("6. Print Statistics");
21             System.out.println("7. Import");
22             System.out.println("8. Export");
23             System.out.println("9. Exit");
24             System.out.print("Enter the option: ");
25
26             // Get option from user
27             int option = 0;
28             try {
29                 option = Integer.parseInt(in.next());
30             } catch (NumberFormatException e) {
31                 System.out.println("Invalid option. Please enter valid option");
32                 continue;
33             }
34             switch (option) {
35                 case 1:
36                     addAccount();
37                     System.out.println("Account has been added successfully!");
38                     break;
39                 case 2:
40                     viewAllAccounts();
41                     break;
42                 case 3:
43                     System.out.print("Enter the Account Id: ");
44                     int id = in.nextInt();
45                     Account acc = null;
46                     try {
47                         acc = accountService.getAccount(id);
48                     } catch (Exception e) {
49                         System.out.println("Account not found!");
50                     }
51                     if (acc != null) {
52                         System.out.println("Account Name: " + acc.getName());
53                         System.out.println("Account Type: " + acc.getType());
54                         System.out.println("Account Balance: " + acc.getBalance());
55                         System.out.println("Is Account Active(Enter true or false): " + acc.isActive());
56                     }
57                     break;
58                 case 4:
59                     updateAccount();
60                     break;
61                 case 5:
62                     deleteAccount();
63                     break;
64                 case 6:
65                     printStatistics();
66                     break;
67                 case 7:
68                     importAccounts();
69                     break;
70                 case 8:
71                     exportAccounts();
72                     break;
73                 case 9:
74                     System.out.println("Exiting...");
75                     break;
76             }
77         }
78     }
79
80     private void addAccount() {
81         System.out.print("Enter Account Name: ");
82         String name = in.nextLine();
83         System.out.print("Enter Account Type: ");
84         String type = in.nextLine();
85         System.out.print("Enter Account Balance: ");
86         double balance = 0;
87         try {
88             balance = Double.parseDouble(in.nextLine());
89         } catch (NumberFormatException e) {
90             System.out.println("Invalid balance. Please enter valid balance");
91             return;
92         }
93         System.out.print("Is Account Active(Enter true or false): ");
94         boolean active = false;
95         try {
96             active = Boolean.parseBoolean(in.nextLine());
97         } catch (Exception e) {
98             System.out.println("Invalid active status. Please enter true or false");
99             return;
100         }
101         Account acc = new Account(name, type, balance, active);
102         accountService.addAccount(acc);
103     }
104
105     private void viewAllAccounts() {
106         List<Account> accounts = accountService.getAllAccounts();
107         if (accounts.isEmpty()) {
108             System.out.println("No accounts found!");
109         } else {
110             System.out.println("All Accounts:");
111             for (Account acc : accounts) {
112                 System.out.println("Id: " + acc.getId() + " Name: " + acc.getName() + " Type: " + acc.getType() + " Balance: " + acc.getBalance() + " Is Active: " + acc.isActive());
113             }
114         }
115     }
116
117     private void viewAccount() {
118         System.out.print("Enter the Account Id: ");
119         int id = in.nextInt();
120         Account acc = null;
121         try {
122             acc = accountService.getAccount(id);
123         } catch (Exception e) {
124             System.out.println("Account not found!");
125         }
126         if (acc != null) {
127             System.out.println("Account Name: " + acc.getName());
128             System.out.println("Account Type: " + acc.getType());
129             System.out.println("Account Balance: " + acc.getBalance());
130             System.out.println("Is Account Active(Enter true or false): " + acc.isActive());
131         }
132     }
133
134     private void updateAccount() {
135         System.out.print("Enter the Account Id to be updated: ");
136         int id = in.nextInt();
137         System.out.print("Enter Account Name: ");
138         String name = in.nextLine();
139         System.out.print("Enter Account Type: ");
140         String type = in.nextLine();
141         System.out.print("Enter Account Balance: ");
142         double balance = 0;
143         try {
144             balance = Double.parseDouble(in.nextLine());
145         } catch (NumberFormatException e) {
146             System.out.println("Invalid balance. Please enter valid balance");
147             return;
148         }
149         System.out.print("Is Account Active(Enter true or false): ");
150         boolean active = false;
151         try {
152             active = Boolean.parseBoolean(in.nextLine());
153         } catch (Exception e) {
154             System.out.println("Invalid active status. Please enter true or false");
155             return;
156         }
157         Account acc = new Account(name, type, balance, active);
158         accountService.updateAccount(id, acc);
159     }
160
161     private void deleteAccount() {
162         System.out.print("Enter the Account Id to be deleted: ");
163         int id = in.nextInt();
164         Account acc = null;
165         try {
166             acc = accountService.getAccount(id);
167         } catch (Exception e) {
168             System.out.println("Account not found!");
169         }
170         if (acc != null) {
171             accountService.deleteAccount(id);
172             System.out.println("Account deleted successfully!");
173         } else {
174             System.out.println("Account not found!");
175         }
176     }
177
178     private void printStatistics() {
179         List<Account> accounts = accountService.getAllAccounts();
180         if (accounts.isEmpty()) {
181             System.out.println("No accounts found!");
182         } else {
183             System.out.println("Account Statistics:");
184             int totalAccounts = 0;
185             double totalBalance = 0;
186             int totalActiveAccounts = 0;
187             for (Account acc : accounts) {
188                 totalAccounts++;
189                 totalBalance += acc.getBalance();
190                 if (acc.isActive()) {
191                     totalActiveAccounts++;
192                 }
193             }
194             System.out.println("Total Accounts: " + totalAccounts);
195             System.out.println("Total Balance: " + totalBalance);
196             System.out.println("Total Active Accounts: " + totalActiveAccounts);
197         }
198     }
199
200     private void importAccounts() {
201         System.out.print("Enter the file path for import: ");
202         String path = in.nextLine();
203         List<Account> accounts = accountService.importAccounts(path);
204         if (accounts.isEmpty()) {
205             System.out.println("No accounts found in the file!");
206         } else {
207             System.out.println("Accounts imported successfully!");
208         }
209     }
210
211     private void exportAccounts() {
212         System.out.print("Enter the file path for export: ");
213         String path = in.nextLine();
214         List<Account> accounts = accountService.exportAccounts(path);
215         if (accounts.isEmpty()) {
216             System.out.println("No accounts found for export!");
217         } else {
218             System.out.println("Accounts exported successfully!");
219         }
220     }
221 }
```

```
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 1
Enter Account Name: Simi
Enter Account Type: Savings
Enter Account Balance: 200000
Is Account Active(Enter true or false): true
Account has been added successfully!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 1
Enter Account Name: Shiny
Enter Account Type: Savings
Enter Account Balance: 900000
Is Account Active(Enter true or false): true
Account has been added successfully!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2
```

## 2. View All Accounts, View Account & Update Account

The screenshot shows the BankingApp.java file with the main method and a while loop for the menu. The console output shows the program running and the user viewing all accounts, viewing a specific account, and updating an account.

```
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
5 public class BankingApp {
6     private static Scanner in;
7     private static AccountService accountService;
8
9     public static void main(String[] args) {
10         in = new Scanner(System.in);
11         accountService = new AccountServiceHashMapImpl();
12         System.out.println("Welcome to Banking Application!");
13         while (true) {
14             System.out.println("\n");
15             System.out.println("1. Add Accounts");
16             System.out.println("2. View All Accounts");
17             System.out.println("3. View Account");
18             System.out.println("4. Update Account");
19             System.out.println("5. Delete Account");
20             System.out.println("6. Print Statistics");
21             System.out.println("7. Import");
22             System.out.println("8. Export");
23             System.out.println("9. Exit");
24             System.out.print("Enter the option: ");
25
26             // Get option from user
27             int option = 0;
28             try {
29                 option = Integer.parseInt(in.next());
30             } catch (NumberFormatException e) {
31                 System.out.println("Invalid option. Please enter valid option");
32                 continue;
33             }
34             switch (option) {
35                 case 1:
36                     addAccount();
37                     System.out.println("Account has been added successfully!");
38                     break;
39                 case 2:
40                     viewAllAccounts();
41                     break;
42                 case 3:
43                     System.out.print("Enter the Account Id: ");
44                     int id = in.nextInt();
45                     Account acc = null;
46                     try {
47                         acc = accountService.getAccount(id);
48                     } catch (Exception e) {
49                         System.out.println("Account not found!");
50                     }
51                     if (acc != null) {
52                         System.out.println("Account Name: " + acc.getName());
53                         System.out.println("Account Type: " + acc.getType());
54                         System.out.println("Account Balance: " + acc.getBalance());
55                         System.out.println("Is Account Active(Enter true or false): " + acc.isActive());
56                     }
57                     break;
58                 case 4:
59                     updateAccount();
60                     break;
61                 case 5:
62                     deleteAccount();
63                     break;
64                 case 6:
65                     printStatistics();
66                     break;
67                 case 7:
68                     importAccounts();
69                     break;
70                 case 8:
71                     exportAccounts();
72                     break;
73                 case 9:
74                     System.out.println("Exiting...");
75                     break;
76             }
77         }
78     }
79
80     private void addAccount() {
81         System.out.print("Enter Account Name: ");
82         String name = in.nextLine();
83         System.out.print("Enter Account Type: ");
84         String type = in.nextLine();
85         System.out.print("Enter Account Balance: ");
86         double balance = 0;
87         try {
88             balance = Double.parseDouble(in.nextLine());
89         } catch (NumberFormatException e) {
90             System.out.println("Invalid balance. Please enter valid balance");
91             return;
92         }
93         System.out.print("Is Account Active(Enter true or false): ");
94         boolean active = false;
95         try {
96             active = Boolean.parseBoolean(in.nextLine());
97         } catch (Exception e) {
98             System.out.println("Invalid active status. Please enter true or false");
99             return;
100         }
101         Account acc = new Account(name, type, balance, active);
102         accountService.addAccount(acc);
103     }
104
105     private void viewAllAccounts() {
106         List<Account> accounts = accountService.getAllAccounts();
107         if (accounts.isEmpty()) {
108             System.out.println("No accounts found!");
109         } else {
110             System.out.println("All Accounts:");
111             for (Account acc : accounts) {
112                 System.out.println("Id: " + acc.getId() + " Name: " + acc.getName() + " Type: " + acc.getType() + " Balance: " + acc.getBalance() + " Is Active: " + acc.isActive());
113             }
114         }
115     }
116
117     private void viewAccount() {
118         System.out.print("Enter the Account Id: ");
119         int id = in.nextInt();
120         Account acc = null;
121         try {
122             acc = accountService.getAccount(id);
123         } catch (Exception e) {
124             System.out.println("Account not found!");
125         }
126         if (acc != null) {
127             System.out.println("Account Name: " + acc.getName());
128             System.out.println("Account Type: " + acc.getType());
129             System.out.println("Account Balance: " + acc.getBalance());
130             System.out.println("Is Account Active(Enter true or false): " + acc.isActive());
131         }
132     }
133
134     private void updateAccount() {
135         System.out.print("Enter the Account Id to be updated: ");
136         int id = in.nextInt();
137         System.out.print("Enter Account Name: ");
138         String name = in.nextLine();
139         System.out.print("Enter Account Type: ");
140         String type = in.nextLine();
141         System.out.print("Enter Account Balance: ");
142         double balance = 0;
143         try {
144             balance = Double.parseDouble(in.nextLine());
145         } catch (NumberFormatException e) {
146             System.out.println("Invalid balance. Please enter valid balance");
147             return;
148         }
149         System.out.print("Is Account Active(Enter true or false): ");
150         boolean active = false;
151         try {
152             active = Boolean.parseBoolean(in.nextLine());
153         } catch (Exception e) {
154             System.out.println("Invalid active status. Please enter true or false");
155             return;
156         }
157         Account acc = new Account(name, type, balance, active);
158         accountService.updateAccount(id, acc);
159     }
160
161     private void deleteAccount() {
162         System.out.print("Enter the Account Id to be deleted: ");
163         int id = in.nextInt();
164         Account acc = null;
165         try {
166             acc = accountService.getAccount(id);
167         } catch (Exception e) {
168             System.out.println("Account not found!");
169         }
170         if (acc != null) {
171             accountService.deleteAccount(id);
172             System.out.println("Account deleted successfully!");
173         } else {
174             System.out.println("Account not found!");
175         }
176     }
177
178     private void printStatistics() {
179         List<Account> accounts = accountService.getAllAccounts();
180         if (accounts.isEmpty()) {
181             System.out.println("No accounts found!");
182         } else {
183             System.out.println("Account Statistics:");
184             int totalAccounts = 0;
185             double totalBalance = 0;
186             int totalActiveAccounts = 0;
187             for (Account acc : accounts) {
188                 totalAccounts++;
189                 totalBalance += acc.getBalance();
190                 if (acc.isActive()) {
191                     totalActiveAccounts++;
192                 }
193             }
194             System.out.println("Total Accounts: " + totalAccounts);
195             System.out.println("Total Balance: " + totalBalance);
196             System.out.println("Total Active Accounts: " + totalActiveAccounts);
197         }
198     }
199
200     private void importAccounts() {
201         System.out.print("Enter the file path for import: ");
202         String path = in.nextLine();
203         List<Account> accounts = accountService.importAccounts(path);
204         if (accounts.isEmpty()) {
205             System.out.println("No accounts found in the file!");
206         } else {
207             System.out.println("Accounts imported successfully!");
208         }
209     }
210
211     private void exportAccounts() {
212         System.out.print("Enter the file path for export: ");
213         String path = in.nextLine();
214         List<Account> accounts = accountService.exportAccounts(path);
215         if (accounts.isEmpty()) {
216             System.out.println("No accounts found for export!");
217         } else {
218             System.out.println("Accounts exported successfully!");
219         }
220     }
221 }
```

```
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2

Id      Name  Type      Balance      Is Active
1       Simi  Savings   200000.0     true
2       Shiny Savings   900000.0     true

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 3
Enter the Account Id: 2

Id      Name  Type      Balance      Is Active
2       Shiny Savings   900000.0     true

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 4
Enter the Account Id to be updated: 2
Enter Account Name: Simu
Enter Account Type: Savings
Enter Account Balance: 700000
Is Account Active(Enter true or false): true
Account has been updated successfully!
```

### 3. Delete Account

```
BankingApp.java x
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
5 public class BankingApp {
6     private static Scanner in;
7     private static AccountService accountService;
8
9     public static void main(String[] args) {
10         in = new Scanner(System.in);
11         accountService = new AccountServiceHashMapImpl();
12         System.out.print("Welcome to Banking Application!");
13         while (true) {
14             System.out.println("\n");
15             System.out.println("1. Add Accounts");
16             System.out.println("2. View All Accounts");
17             System.out.println("3. View Account");
18             System.out.println("4. Update Account");
19             System.out.println("5. Delete Account");
20             System.out.println("6. Print Statistics");
21             System.out.println("7. Import");
22             System.out.println("8. Export");
23             System.out.println("9. Exit");
24             System.out.print("Enter the option: ");
25
26             // Get option from user
27             int option = 0;
28             try {
29                 option = Integer.parseInt(in.next());
30             } catch (NumberFormatException e) {
31                 System.out.println("Invalid option. Please enter valid option");
32                 continue;
33             }
34             switch (option) {
35                 case 1:
36                     addAccount();
37                     System.out.println("Account has been added successfully!");
38                     break;
39                 case 2:
40                     viewAllAccounts();
41                     break;
42                 case 3:
43                     System.out.print("Enter the Account Id: ");
44                     int id = in.nextInt();
45                     Account acc = null;
46
47 BankingApp (6) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.just.openjdk.hotspot
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2


| Id | Name | Type    | Balance  | Is Active |
|----|------|---------|----------|-----------|
| 1  | Simi | Savings | 200000.0 | true      |
| 2  | Simu | Savings | 700000.0 | true      |


1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 5
Enter the Account Id to be deleted: 1
Account has been deleted successfully!
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2


| Id | Name | Type    | Balance  | Is Active |
|----|------|---------|----------|-----------|
| 2  | Simu | Savings | 700000.0 | true      |


1. Add Accounts
2. View All Accounts
3. View Account
```

The screenshot displays an IDE with two panels. The left panel shows the source code for `BankingApp.java`, and the right panel shows the console output.

**Source Code (BankingApp.java):**

```

1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
5 public class BankingApp {
6     private static Scanner in;
7     private static AccountService accountService;
8
9     public static void main(String[] args) {
10         in = new Scanner(System.in);
11         accountService = new AccountServiceHashMapImpl();
12         System.out.print("Welcome to Banking Application!");
13         while (true) {
14             System.out.println("\n");
15             System.out.println("1. Add Accounts");
16             System.out.println("2. View All Accounts");
17             System.out.println("3. View Account");
18             System.out.println("4. Update Account");
19             System.out.println("5. Delete Account");
20             System.out.println("6. Print Statistics");
21             System.out.println("7. Import");
22             System.out.println("8. Export");
23             System.out.println("9. Exit");
24             System.out.print("Enter the option: ");
25
26             // Get option from user
27             int option = 0;
28             try {
29                 option = Integer.parseInt(in.next());
30             } catch (NumberFormatException e) {
31                 System.out.println("Invalid option. Please enter valid option");
32                 continue;
33             }
34             try {
35                 switch (option) {
36                     case 1:
37                         addAccount();
38                         System.out.println("Account has been added successfully!");
39                         break;
40                     case 2:
41                         viewAllAccounts();
42                         break;
43                     case 3:
44                         System.out.print("Enter the Account Id: ");
45                         int id = in.nextInt();
46                         Account acc = null;

```

**Console Output:**

```

BankingApp [6] [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot...
Enter the option: 1
Enter the Account Id to be deleted: 1
Account has been deleted successfully!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2

   Id      Name Type      Balance      Is Active
   --      -  -  -
    2      Simu Savings  700000.0      true

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 5
Enter the Account Id to be deleted: 2
Account has been deleted successfully!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2

   Id      Name Type      Balance      Is Active
   --      -  -  -

```

## 4. Import & Export

```
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
11 public class BankingApp {
12     private static Scanner in;
13     private static AccountService accountService;
14
15
16 public static void main(String[] args) {
17     in = new Scanner(System.in);
18     accountService = new AccountServiceHashMapImpl();
19     System.out.print("Welcome to Banking Application!");
20     while (true) {
21         System.out.println("\n");
22         System.out.println("1. Add Accounts");
23         System.out.println("2. View All Accounts");
24         System.out.println("3. View Account");
25         System.out.println("4. Update Account");
26         System.out.println("5. Delete Account");
27         System.out.println("6. Print Statistics");
28         System.out.println("7. Import");
29         System.out.println("8. Export");
30         System.out.println("9. Exit");
31         System.out.print("Enter the option: ");
32
33         // Get option from user
34         int option = 0;
35         try {
36             option = Integer.parseInt(in.next());
37         } catch (NumberFormatException e) {
38             System.out.println("Invalid option. Please enter valid option");
39             continue;
40         }
41         try {
42             switch (option) {
43
44                 case 1:
45                     addAccount();
46                     System.out.println("Account has been added successfully!");
47                     break;
48                 case 2:
49                     viewAllAccounts();
50                     break;
51                 case 3:
52                     System.out.print("Enter the Account Id: ");
53                     int id = in.nextInt();
54                     Account acc = null;
55                     try {
```

BankingApp (6) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86\_64\plugins\org.eclipse.justi.openjdk.hotspot

1. Add Accounts  
2. View All Accounts  
3. View Account  
4. Update Account  
5. Delete Account  
6. Print Statistics  
7. Import  
8. Export  
9. Exit  
Enter the option: 7

- Import started  
Importing file...  
6 Accounts are imported successfully.

Id	Name	Type	Balance	Is Active
1	Stella	Savings	100000.0	true
2	Steffy	Savings	200000.0	true
3	Hadi	Savings	300000.0	true
4	Juli	Current	400000.0	true
5	Eliza	Current	500000.0	true
6	Sam	Current	600000.0	true

1. Add Accounts  
2. View All Accounts  
3. View Account  
4. Update Account  
5. Delete Account  
6. Print Statistics  
7. Import  
8. Export  
9. Exit  
Enter the option: 2

- Export started  
6 Accounts are exported successfully.

```
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
11 public class BankingApp {
12     private static Scanner in;
13     private static AccountService accountService;
14
15
16 public static void main(String[] args) {
17     in = new Scanner(System.in);
18     accountService = new AccountServiceHashMapImpl();
19     System.out.print("Welcome to Banking Application!");
20     while (true) {
21         System.out.println("\n");
22         System.out.println("1. Add Accounts");
23         System.out.println("2. View All Accounts");
24         System.out.println("3. View Account");
25         System.out.println("4. Update Account");
26         System.out.println("5. Delete Account");
27         System.out.println("6. Print Statistics");
28         System.out.println("7. Import");
29         System.out.println("8. Export");
30         System.out.println("9. Exit");
31         System.out.print("Enter the option: ");
32
33         // Get option from user
34         int option = 0;
35         try {
36             option = Integer.parseInt(in.next());
37         } catch (NumberFormatException e) {
38             System.out.println("Invalid option. Please enter valid option");
39             continue;
40         }
41         try {
42             switch (option) {
43
44                 case 1:
45                     addAccount();
46                     System.out.println("Account has been added successfully!");
47                     break;
48                 case 2:
49                     viewAllAccounts();
50                     break;
51                 case 3:
52                     System.out.print("Enter the Account Id: ");
53                     int id = in.nextInt();
54                     Account acc = null;
55                     try {
```

1,Stella,Savings,100000,true  
2,Steffy,Savings,200000,true  
3,3,Hadi,Savings,300000,true  
4,4,Juli,Current,400000,true  
5,5,Eliza,Current,500000,true  
6,6,Sam,Current,600000,true

```
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
11 public class BankingApp {
12     private static Scanner in;
13     private static AccountService accountService;
14
15
16 public static void main(String[] args) {
17     in = new Scanner(System.in);
18     accountService = new AccountServiceHashMapImpl();
19     System.out.print("Welcome to Banking Application!");
20     while (true) {
21         System.out.println("\n");
22         System.out.println("1. Add Accounts");
23         System.out.println("2. View All Accounts");
24         System.out.println("3. View Account");
25         System.out.println("4. Update Account");
26         System.out.println("5. Delete Account");
27         System.out.println("6. Print Statistics");
28         System.out.println("7. Import");
29         System.out.println("8. Export");
30         System.out.println("9. Exit");
31         System.out.print("Enter the option: ");
32
33         // Get option from user
34         int option = 0;
35         try {
36             option = Integer.parseInt(in.next());
37         } catch (NumberFormatException e) {
38             System.out.println("Invalid option. Please enter valid option");
39             continue;
40         }
41         try {
42             switch (option) {
43
44                 case 1:
45                     addAccount();
46                     System.out.println("Account has been added successfully!");
47                     break;
48                 case 2:
49                     viewAllAccounts();
50                     break;
51                 case 3:
52                     System.out.print("Enter the Account Id: ");
53                     int id = in.nextInt();
54                     Account acc = null;
55                     try {
```

1,Stella,Savings,100000.0,true  
2,2,Steffy,Savings,200000.0,true  
3,3,Hadi,Savings,300000.0,true  
4,4,Juli,Current,400000.0,true  
5,5,Eliza,Current,500000.0,true  
6,6,Sam,Current,600000.0,true  
7

## 5. Print Statistics & Exit

```
BankingApp.java
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
11
12 public class BankingApp {
13     private static Scanner in;
14     private static AccountService accountService;
15
16 public static void main(String[] args) {
17     in = new Scanner(System.in);
18     accountService = new AccountServiceHashMapImpl();
19     System.out.print("Welcome to Banking Application!");
20     while (true) {
21         System.out.println("\n");
22         System.out.println("1. Add Accounts");
23         System.out.println("2. View All Accounts");
24         System.out.println("3. View Account");
25         System.out.println("4. Update Account");
26         System.out.println("5. Delete Account");
27         System.out.println("6. Print Statistics");
28         System.out.println("7. Import");
29         System.out.println("8. Export");
30         System.out.println("9. Exit");
31         System.out.print("Enter the option: ");
32
33         // Get option from user
34         int option = 0;
35         try {
36             option = Integer.parseInt(in.next());
37         } catch (NumberFormatException e) {
38             System.out.println("Invalid option. Please enter valid option");
39             continue;
40         }
41         try {
42             switch (option) {
43
44                 case 1:
45                     addAccount();
46                     System.out.println("Account has been added successfully!");
47             }
48         }
49     }
50 }
```

```
Console
BankingApp (6) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justi.openjdk.hotspot
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 6
a) No of accounts which has balance more than 1 lac
b) Show no of account by account type
c) Show no of accounts by account type with sorting
d) Show avg balance by account type
e) List account ids whose account name contains given name
Enter an option: a
a) No of accounts which has balance more than 1 lac: 5
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 6
a) No of accounts which has balance more than 1 lac
b) Show no of account by account type
c) Show no of accounts by account type with sorting
d) Show avg balance by account type
e) List account ids whose account name contains given name
Enter an option: b
b) Show no of account by account type: {Savings=3, Current=3}
```

```
BankingApp.java
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
11
12 public class BankingApp {
13     private static Scanner in;
14     private static AccountService accountService;
15
16 public static void main(String[] args) {
17     in = new Scanner(System.in);
18     accountService = new AccountServiceHashMapImpl();
19     System.out.print("Welcome to Banking Application!");
20     while (true) {
21         System.out.println("\n");
22         System.out.println("1. Add Accounts");
23         System.out.println("2. View All Accounts");
24         System.out.println("3. View Account");
25         System.out.println("4. Update Account");
26         System.out.println("5. Delete Account");
27         System.out.println("6. Print Statistics");
28         System.out.println("7. Import");
29         System.out.println("8. Export");
30         System.out.println("9. Exit");
31         System.out.print("Enter the option: ");
32
33         // Get option from user
34         int option = 0;
35         try {
36             option = Integer.parseInt(in.next());
37         } catch (NumberFormatException e) {
38             System.out.println("Invalid option. Please enter valid option");
39             continue;
40         }
41         try {
42             switch (option) {
43
44                 case 1:
45                     addAccount();
46                     System.out.println("Account has been added successfully!");
47             }
48         }
49     }
50 }
```

```
Console
BankingApp (6) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justi.openjdk.hotspot
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 6
a) No of accounts which has balance more than 1 lac
b) Show no of account by account type
c) Show no of accounts by account type with sorting
d) Show avg balance by account type
e) List account ids whose account name contains given name
Enter an option: c
c) Show no of accounts by account type with sorting: {Current=3, Savings=3}
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 6
a) No of accounts which has balance more than 1 lac
b) Show no of account by account type
c) Show no of accounts by account type with sorting
d) Show avg balance by account type
e) List account ids whose account name contains given name
Enter an option: d
d) Show avg balance by account type: {Savings=200000.0, Current=500000.0}
```

```
BankingApp.java ×
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
11 public class BankingApp {
12     private static Scanner in;
13     private static AccountService accountService;
14
15     public static void main(String[] args) {
16         in = new Scanner(System.in);
17         accountService = new AccountServiceHashMapImpl();
18         System.out.print("Welcome to Banking Application!");
19         while (true) {
20             System.out.println("\n");
21             System.out.println("1. Add Accounts");
22             System.out.println("2. View All Accounts");
23             System.out.println("3. View Account");
24             System.out.println("4. Update Account");
25             System.out.println("5. Delete Account");
26             System.out.println("6. Print Statistics");
27             System.out.println("7. Import");
28             System.out.println("8. Export");
29             System.out.println("9. Exit");
30             System.out.print("Enter the option: ");
31
32             // Get option from user
33             int option = 0;
34             try {
35                 option = Integer.parseInt(in.next());
36             } catch (NumberFormatException e) {
37                 System.out.println("Invalid option. Please enter valid option");
38                 continue;
39             }
40             try {
41                 switch (option) {
42                     case 1:
43                         addAccount();
44                         System.out.println("Account has been added successfully!");
45                         break;
46                     case 2:
47                         viewAllAccounts();
48                         break;
49                     case 3:
50                         System.out.print("Enter the Account Id: ");
51                         int id = in.nextInt();
52                         Account acc = null;
53                         try {
54                             acc = accountService.getAccount(id);
55                             if (acc != null) {
56                                 System.out.println("Account found: " + acc);
57                             } else {
58                                 System.out.println("Account not found");
59                             }
60                         } catch (Exception e) {
61                             System.out.println("Error: " + e.getMessage());
62                         }
63                     case 4:
64                         System.out.print("Enter the Account Id: ");
65                         int id = in.nextInt();
66                         Account acc = null;
67                         try {
68                             acc = accountService.getAccount(id);
69                             if (acc != null) {
70                                 System.out.println("Account found: " + acc);
71                             } else {
72                                 System.out.println("Account not found");
73                             }
74                         } catch (Exception e) {
75                             System.out.println("Error: " + e.getMessage());
76                         }
77                     case 5:
78                         deleteAccount();
79                         System.out.println("Account deleted successfully!");
80                         break;
81                     case 6:
82                         printStatistics();
83                         break;
84                     case 7:
85                         importAccounts();
86                         break;
87                     case 8:
88                         exportAccounts();
89                         break;
90                     case 9:
91                         System.out.println("Thank you!!!");
92                         break;
93                     default:
94                         System.out.println("Invalid option. Please enter valid option");
95                         continue;
96                 }
97             }
98         }
99     }
100 }
```

Console × account-input.txt account-output.txt

BankingApp (6) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86\_64\plugins\org.eclipse.just.openjdk.hotspot...  
Enter an option: 6  
d) Show avg balance by account type: {Savings=200000.0, Current=500000.0}

1. Add Accounts  
2. View All Accounts  
3. View Account  
4. Update Account  
5. Delete Account  
6. Print Statistics  
7. Import  
8. Export  
9. Exit  
Enter the option: 6  
a) No of accounts which has balance more than 1 lac  
b) Show no of account by account type  
c) Show no of accounts by account type with sorting  
d) Show avg balance by account type  
e) List account ids whose account name contains given name  
Enter an option: e  
Enter the name: Ste  
e) List account ids whose account name contains given name: [1, 2]

1. Add Accounts  
2. View All Accounts  
3. View Account  
4. Update Account  
5. Delete Account  
6. Print Statistics  
7. Import  
8. Export  
9. Exit  
Enter the option: 9  
Thank you!!!

1. Add Accounts  
2. View All Accounts  
3. View Account  
4. Update Account  
5. Delete Account  
6. Print Statistics  
7. Import  
8. Export  
9. Exit  
Enter the option: