

Banking App Program using ArrayList Screenshots

1. Add Accounts & View All Accounts

The screenshot shows the BankingApp.java file and the console output for the first step of the program. The code defines a BankingApp class with a main method that uses a Scanner to take user input. It has a switch statement with cases for adding, viewing, updating, and deleting accounts. The console output shows the program running and the user adding an account named Stella with a balance of 1234.0 and an active status of true.

```
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 AccountServiceArrayListImpl();
12         // accountService = new AccountServiceHashMapImpl();
13         // accountService = new AccountServiceHashSetImpl();
14         // accountService = new AccountServiceLinkedListImpl();
15         // accountService = new AccountServiceTreeMapImpl();
16         // accountService = new AccountServiceTreeSetImpl();
17         System.out.print("Welcome to Banking Application!");
18         while (true) {
19             System.out.println("\n");
20             System.out.println("1. Add Accounts");
21             System.out.println("2. View All Accounts");
22             System.out.println("3. View Account");
23             System.out.println("4. Update Account");
24             System.out.println("5. Delete Account");
25             System.out.println("6. Exit");
26             System.out.print("Enter the option: ");
27
28             int option = 0;
29             option = Integer.parseInt(in.next());
30             switch (option) {
31                 case 1:
32                     addAccount();
33                     System.out.println("Account has been added successfully");
34                     break;
35                 case 2:
36                     viewAllAccounts();
37                     break;
38                 case 3:
39                     System.out.print("Enter the Account Id: ");
40                     int id = in.nextInt();
41                     Account acc = viewAccount(id);
42                     printHeader();
43                     printDetail(acc);
44                     break;
45                 case 4:
46                     updateAccount();
47                     System.out.println("Account has been updated successfully");
48                     break;
49                 case 5:
50                     deleteAccount();
51                     break;
52                 case 6:
53                     System.out.println("Exit");
54                     break;
55             }
56         }
57     }
58
59     private void addAccount() {
60         System.out.print("Enter Account Name: ");
61         String name = in.next();
62         System.out.print("Enter Account Type: ");
63         String type = in.next();
64         System.out.print("Enter Account Balance: ");
65         double balance = in.nextDouble();
66         System.out.print("Is Account Active(Enter true or false): ");
67         boolean active = in.nextBoolean();
68         accountService.addAccount(new Account(name, type, balance, active));
69         System.out.println("Account has been added successfully!");
70     }
71
72     private void viewAllAccounts() {
73         System.out.println("\n");
74         System.out.println("Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active");
75         for (Account acc : accountService.getAllAccounts()) {
76             System.out.println(acc.getId() + "\t\t\t\t\t" + acc.getName() + "\t\t\t\t\t" + acc.getType() + "\t\t\t\t\t" + acc.getBalance() + "\t\t\t\t\t" + acc.isActive());
77         }
78     }
79
80     private void viewAccount(int id) {
81         Account acc = accountService.getAccount(id);
82         printHeader();
83         printDetail(acc);
84     }
85
86     private void printHeader() {
87         System.out.println("\n");
88         System.out.println("Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active");
89     }
90
91     private void printDetail(Account acc) {
92         System.out.println("\n");
93         System.out.println("Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active");
94         System.out.println(acc.getId() + "\t\t\t\t\t" + acc.getName() + "\t\t\t\t\t" + acc.getType() + "\t\t\t\t\t" + acc.getBalance() + "\t\t\t\t\t" + acc.isActive());
95     }
96
97     private void updateAccount() {
98         System.out.print("Enter the Account Id to be updated: ");
99         int id = in.nextInt();
100         Account acc = viewAccount(id);
101         System.out.print("Enter Account Name: ");
102         String name = in.next();
103         System.out.print("Enter Account Type: ");
104         String type = in.next();
105         System.out.print("Enter Account Balance: ");
106         double balance = in.nextDouble();
107         System.out.print("Is Account Active(Enter true or false): ");
108         boolean active = in.nextBoolean();
109         accountService.updateAccount(id, new Account(name, type, balance, active));
110         System.out.println("Account has been updated successfully!");
111     }
112
113     private void deleteAccount() {
114         System.out.print("Enter the Account Id to be deleted: ");
115         int id = in.nextInt();
116         accountService.deleteAccount(id);
117         System.out.println("Account has been deleted successfully!");
118     }
119 }
```

Console Output:

```
BankingApp [4] [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.justi.openjdk
Welcome to Banking Application!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Exit
Enter the option: 1
Enter Account Name: Stella
Enter Account Type: Savings
Enter Account Balance: 1234
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. Exit
Enter the option: 2

Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active
1\t\t\t\t\tStella\t\t\t\t\tSavings\t\t\t\t\t1234.0\t\t\t\t\ttrue
```

2. View Account & Update Account

The screenshot shows the BankingApp.java file and the console output for the second step of the program. The code is the same as in the first screenshot, but the console output shows the user viewing and updating an account. The user enters option 3 to view an account, then option 4 to update it. The console output shows the account details for Stella and then the updated details for Had1.

```
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 AccountServiceArrayListImpl();
12         // accountService = new AccountServiceHashMapImpl();
13         // accountService = new AccountServiceHashSetImpl();
14         // accountService = new AccountServiceLinkedListImpl();
15         // accountService = new AccountServiceTreeMapImpl();
16         // accountService = new AccountServiceTreeSetImpl();
17         System.out.print("Welcome to Banking Application!");
18         while (true) {
19             System.out.println("\n");
20             System.out.println("1. Add Accounts");
21             System.out.println("2. View All Accounts");
22             System.out.println("3. View Account");
23             System.out.println("4. Update Account");
24             System.out.println("5. Delete Account");
25             System.out.println("6. Exit");
26             System.out.print("Enter the option: ");
27
28             int option = 0;
29             option = Integer.parseInt(in.next());
30             switch (option) {
31                 case 1:
32                     addAccount();
33                     System.out.println("Account has been added successfully");
34                     break;
35                 case 2:
36                     viewAllAccounts();
37                     break;
38                 case 3:
39                     System.out.print("Enter the Account Id: ");
40                     int id = in.nextInt();
41                     Account acc = viewAccount(id);
42                     printHeader();
43                     printDetail(acc);
44                     break;
45                 case 4:
46                     updateAccount();
47                     System.out.println("Account has been updated successfully");
48                     break;
49                 case 5:
50                     deleteAccount();
51                     break;
52                 case 6:
53                     System.out.println("Exit");
54                     break;
55             }
56         }
57     }
58
59     private void addAccount() {
60         System.out.print("Enter Account Name: ");
61         String name = in.next();
62         System.out.print("Enter Account Type: ");
63         String type = in.next();
64         System.out.print("Enter Account Balance: ");
65         double balance = in.nextDouble();
66         System.out.print("Is Account Active(Enter true or false): ");
67         boolean active = in.nextBoolean();
68         accountService.addAccount(new Account(name, type, balance, active));
69         System.out.println("Account has been added successfully!");
70     }
71
72     private void viewAllAccounts() {
73         System.out.println("\n");
74         System.out.println("Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active");
75         for (Account acc : accountService.getAllAccounts()) {
76             System.out.println(acc.getId() + "\t\t\t\t\t" + acc.getName() + "\t\t\t\t\t" + acc.getType() + "\t\t\t\t\t" + acc.getBalance() + "\t\t\t\t\t" + acc.isActive());
77         }
78     }
79
80     private void viewAccount(int id) {
81         Account acc = accountService.getAccount(id);
82         printHeader();
83         printDetail(acc);
84     }
85
86     private void printHeader() {
87         System.out.println("\n");
88         System.out.println("Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active");
89     }
90
91     private void printDetail(Account acc) {
92         System.out.println("\n");
93         System.out.println("Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active");
94         System.out.println(acc.getId() + "\t\t\t\t\t" + acc.getName() + "\t\t\t\t\t" + acc.getType() + "\t\t\t\t\t" + acc.getBalance() + "\t\t\t\t\t" + acc.isActive());
95     }
96
97     private void updateAccount() {
98         System.out.print("Enter the Account Id to be updated: ");
99         int id = in.nextInt();
100         Account acc = viewAccount(id);
101         System.out.print("Enter Account Name: ");
102         String name = in.next();
103         System.out.print("Enter Account Type: ");
104         String type = in.next();
105         System.out.print("Enter Account Balance: ");
106         double balance = in.nextDouble();
107         System.out.print("Is Account Active(Enter true or false): ");
108         boolean active = in.nextBoolean();
109         accountService.updateAccount(id, new Account(name, type, balance, active));
110         System.out.println("Account has been updated successfully!");
111     }
112
113     private void deleteAccount() {
114         System.out.print("Enter the Account Id to be deleted: ");
115         int id = in.nextInt();
116         accountService.deleteAccount(id);
117         System.out.println("Account has been deleted successfully!");
118     }
119 }
```

Console Output:

```
BankingApp [4] [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.justi.openj
1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Exit
Enter the option: 2

Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active
1\t\t\t\t\tStella\t\t\t\t\tSavings\t\t\t\t\t1234.0\t\t\t\t\ttrue
2\t\t\t\t\tSteffy\t\t\t\t\tSavings\t\t\t\t\t2343.0\t\t\t\t\ttrue

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Exit
Enter the option: 3
Enter the Account Id: 1

Id\t\t\t\t\tName\t\t\t\t\tType\t\t\t\t\tBalance\t\t\t\t\tIs Active
1\t\t\t\t\tStella\t\t\t\t\tSavings\t\t\t\t\t1234.0\t\t\t\t\ttrue

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Exit
Enter the option: 4
Enter the Account Id to be updated: 2
Enter Account Name: Had1
Enter Account Type: Savings
Enter Account Balance: 12323
Is Account Active(Enter true or false): true
Account has been updated successfully!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Exit
```

3. Delete Account & Exit Option

```
BankingApp.java ×
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
16
17 public class BankingApp {
18     private static Scanner in;
19     private static AccountService accountService;
20
21     public static void main(String[] args) {
22         in = new Scanner(System.in);
23         accountService = new AccountServiceArrListImpl();
24         // accountService = new AccountServiceHashMapImpl();
25         // accountService = new AccountServiceHashSetImpl();
26         // accountService = new AccountServiceLinkedListImpl();
27         // accountService = new AccountServiceTreeMapImpl();
28         // accountService = new AccountServiceTreeSetImpl();
29         System.out.println("Welcome to Banking Application!");
30         while (true) {
31             System.out.println("\n");
32             System.out.println("1. Add Accounts");
33             System.out.println("2. View All Accounts");
34             System.out.println("3. View Account");
35             System.out.println("4. Update Account");
36             System.out.println("5. Delete Account");
37             System.out.println("6. Exit");
38             System.out.print("Enter the option: ");
39
40             int option = 0;
41             option = Integer.parseInt(in.next());
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 = viewAccount(id);
55                     printHeader();
56                     printDetail(acc);
57                     break;
58                 case 4:
59                     updateAccount();
60                     System.out.println("Account has been updated successfully");
61             }
62         }
63     }
64 }
```

```
Console ×
BankingApp (4) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openj
Enter Account Name: Hadi
Enter Account Type: Savings
Enter Account Balance: 12323
Is Account Active(Enter true or false): true
1
Account has been updated successfully!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. 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. Exit
Enter the option: 2

Id      Name  Type      Balance  Is Active
1       Stella Savings  1234.0    true

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Exit
Enter the option: 6
Thank you!!!

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Exit
Enter the option:
```