

Banking App Program using LinkedList Screenshots

1. Add Accounts, View All Accounts, & View Account

The screenshot shows the BankingApp.java file and the console output. The code implements a menu-driven application with options to add, view all, view, update, delete, and exit. The console shows the user selecting option 1 to add an account, then option 2 to view all accounts, and finally option 3 to view a specific 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 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                     System.out.println("Account has been deleted successfully!");
52                     break;
53                 case 6:
54                     System.out.println("Exiting...");
55                     return;
56             }
57         }
58     }
59
60     private void addAccount() {
61         System.out.print("Enter Account Name: ");
62         String name = in.nextLine();
63         System.out.print("Enter Account Type: ");
64         String type = in.nextLine();
65         System.out.print("Enter Account Balance: ");
66         double balance = in.nextDouble();
67         System.out.print("Is Account Active(Enter true or false): ");
68         boolean active = in.nextBoolean();
69         accountService.addAccount(new Account(name, type, balance, active));
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 Account viewAccount(int id) {
81         return accountService.getAccount(id);
82     }
83
84     private void printHeader() {
85         System.out.println("\n");
86         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");
87     }
88
89     private void printDetail(Account acc) {
90         System.out.println("\n");
91         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");
92         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());
93     }
94
95     private void updateAccount() {
96         System.out.print("Enter the Account Id to be updated: ");
97         int id = in.nextInt();
98         System.out.print("Enter Account Name: ");
99         String name = in.nextLine();
100         System.out.print("Enter Account Type: ");
101         String type = in.nextLine();
102         System.out.print("Enter Account Balance: ");
103         double balance = in.nextDouble();
104         System.out.print("Is Account Active(Enter true or false): ");
105         boolean active = in.nextBoolean();
106         accountService.updateAccount(id, new Account(name, type, balance, active));
107     }
108
109     private void deleteAccount() {
110         System.out.print("Enter the Account Id to be deleted: ");
111         int id = in.nextInt();
112         accountService.deleteAccount(id);
113     }
114 }
```

Console Output:

```
BankingApp (4) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\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: 1232
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\t1232.0\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\t1232.0\t\t\t\t\ttrue
```

2. Update Account & Delete Account

The screenshot shows the BankingApp.java file and the console output. The code implements a menu-driven application with options to add, view all, view, update, delete, and exit. The console shows the user selecting option 4 to update an account and then option 5 to delete 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 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                     System.out.println("Account has been deleted successfully!");
52                     break;
53                 case 6:
54                     System.out.println("Exiting...");
55                     return;
56             }
57         }
58     }
59
60     private void addAccount() {
61         System.out.print("Enter Account Name: ");
62         String name = in.nextLine();
63         System.out.print("Enter Account Type: ");
64         String type = in.nextLine();
65         System.out.print("Enter Account Balance: ");
66         double balance = in.nextDouble();
67         System.out.print("Is Account Active(Enter true or false): ");
68         boolean active = in.nextBoolean();
69         accountService.addAccount(new Account(name, type, balance, active));
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 Account viewAccount(int id) {
81         return accountService.getAccount(id);
82     }
83
84     private void printHeader() {
85         System.out.println("\n");
86         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");
87     }
88
89     private void printDetail(Account acc) {
90         System.out.println("\n");
91         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");
92         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());
93     }
94
95     private void updateAccount() {
96         System.out.print("Enter the Account Id to be updated: ");
97         int id = in.nextInt();
98         System.out.print("Enter Account Name: ");
99         String name = in.nextLine();
100         System.out.print("Enter Account Type: ");
101         String type = in.nextLine();
102         System.out.print("Enter Account Balance: ");
103         double balance = in.nextDouble();
104         System.out.print("Is Account Active(Enter true or false): ");
105         boolean active = in.nextBoolean();
106         accountService.updateAccount(id, new Account(name, type, balance, active));
107     }
108
109     private void deleteAccount() {
110         System.out.print("Enter the Account Id to be deleted: ");
111         int id = in.nextInt();
112         accountService.deleteAccount(id);
113     }
114 }
```

Console Output:

```
BankingApp (4) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justi.openjdk
Enter the option: 1
Enter Account Name: Steffy
Enter Account Type: Savings
Enter Account Balance: 1212
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\t1232.0\t\t\t\t\ttrue
2\t\t\t\t\tSteffy\t\t\t\t\tSavings\t\t\t\t\t1212.0\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: 1
Enter Account Name: Hadi
Enter Account Type: Savings
Enter Account Balance: 121
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
Enter the option: 5
Enter the Account Id to be deleted: 2
Account has been deleted successfully!
```

3. Delete Account & Exit Option

```
BankingApp.java
1 package com.indium.bankingapp;
2
3 import java.util.Collection;
4
16 public class BankingApp {
17     private static Scanner in;
18     private static AccountService accountService;
19
20
21     public static void main(String[] args) {
22         in = new Scanner(System.in);
23         // accountService = new AccountServiceArrayListImpl();
24         // accountService = new AccountServiceHashMapImpl();
25         // accountService = new AccountServiceHashSetImpl();
26         accountService = new AccountServiceLinkedListImpl();
27         // accountService = new AccountServiceTreeMapImpl();
28         // accountService = new AccountServiceTreeSetImpl();
29         System.out.print("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.openjdk
Enter Account Name: Hadi
Enter Account Type: Savings
Enter Account Balance: 121
Is Account Active(Enter true or false): true
0
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       Hadi  Savings    121.0        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:
```