

K.R. Mangalam University

School of Engineering & Technology

Assignment 1

Java Programming

Submitted by:

Name: Rakesh G

Roll No: 2401201064

Class: BCA (AI & DS)

Submitted to:

Dr. Manish Kumar

GitHub Repository:

https://github.com/rakesh4407/Java_Assignment_Rakesh

CODE:

```
BankingApp.java 2 •
JAVA > BankingApp.java > Language Support for Java(TM) by Red Hat > Account
1 // RAKESH G
2 // 240120164
3
4 import java.util.Scanner;
5
6 // Account class
7 class Account {
8
9     private int accountNumber;    Field accountNumber can be final
10    private String name, email, phone;    Field name can be final
11    private double balance;
12
13    Account(int accNo, String name, double balance, String email, String phone) {
14        this.accountNumber = accNo;
15        this.name = name;
16        this.balance = balance;
17        this.email = email;
18        this.phone = phone;
19    }
20
21    public int getAccountNumber() {
22        return accountNumber;
23    }
24
25    void deposit(double amt) {
26        if (amt > 0) {
27            balance += amt;
28            System.out.println("Deposit Successful. Balance: " + balance);
29        } else {
30            System.out.println("Invalid deposit amount.");
31        }
32    }
33
34    void withdraw(double amt) {
35        if (amt > 0 && amt <= balance) {
36            balance -= amt;
37            System.out.println("Withdrawal Successful. Balance: " + balance);
38        } else {
```

```
BankingApp.java 2
D:\RAKESH\VS\JAVA\BankingApp.java • 2 problems in this file by Red Hat > Account

7 class Account {
34     void withdraw(double amt) {
37         System.out.println("Withdrawal Successful. Balance: " + balance);
38     } else {
39         System.out.println(x:"Invalid or insufficient balance.");
40     }
41 }
42
43 void show() {
44     System.out.println(accountNumber + " | " + name + " | " + balance + " | " + email + " | " + phone);
45 }
46
47 void update(String email, String phone) {
48     this.email = email;
49     this.phone = phone;
50     System.out.println(x:"Contact updated!");
51 }
52 }
53
54 // Main Banking Application
55 public class BankingApp {
56
57     static Scanner sc = new Scanner(System.in);
58     static Account[] accounts = new Account[100];
59     static int count = 0;
60
61     static Account find(int accNo) {
62         for (int i = 0; i < count; i++) {
63             if (accounts[i].getAccountNumber() == accNo) {
64                 return accounts[i];
65             }
66         }
67         return null;
68     }
69
70     public static void main(String[] args) {
71         while (true) {
72             System.out.println("\n1. Create 2. Deposit 3. Withdraw 4. View 5. Update 6. Exit");
73             int choice = sc.nextInt();
74             switch (choice) {
75                 case 1:
76                     createAccount();
77                     break;
78                 case 2:
79                     deposit();
80                     break;
81                 case 3:
82                     withdraw();
83                     break;
84                 case 4:
85                     view();
86                     break;
87                 case 5:
88                     update();
89                     break;
90                 case 6:
91                     exit();
92                     break;
93                 default:
94                     System.out.println("Invalid choice");
95             }
96         }
97     }
98
99     static void createAccount() {
100         if (count == 100) {
101             System.out.println("Account limit reached");
102             return;
103         }
104         Account acc = new Account();
105         acc.setAccountNumber(count + 1);
106         acc.setName(sc.next());
107         acc.setBalance(0);
108         acc.setEmail(sc.next());
109         acc.setPhone(sc.next());
110         accounts[count] = acc;
111         count++;
112     }
113
114     static void deposit() {
115         int accNo = sc.nextInt();
116         Account acc = find(accNo);
117         if (acc == null) {
118             System.out.println("Account not found");
119             return;
120         }
121         double amt = sc.nextDouble();
122         acc.deposit(amt);
123     }
124
125     static void withdraw() {
126         int accNo = sc.nextInt();
127         Account acc = find(accNo);
128         if (acc == null) {
129             System.out.println("Account not found");
130             return;
131         }
126         double amt = sc.nextDouble();
127         acc.withdraw(amt);
128     }
129
130     static void view() {
131         for (int i = 0; i < count; i++) {
132             accounts[i].show();
133         }
134     }
135
136     static void update() {
137         int accNo = sc.nextInt();
138         Account acc = find(accNo);
139         if (acc == null) {
140             System.out.println("Account not found");
141             return;
142         }
143         String email = sc.next();
144         String phone = sc.next();
145         acc.update(email, phone);
146     }
147
148     static void exit() {
149         System.out.println("Exiting...");
150         System.exit(0);
151     }
152 }
153
154 Run main | Debug main | Run | Debug
155
```

BankingApp.java 2

JAVA > BankingApp.java > Language Support for Java(TM) by Red Hat > Account

```
55 public class BankingApp {
70     public static void main(String[] args) {
73         System.out.print(s:"Enter choice: ");
74         int ch = sc.nextInt();
75         sc.nextLine();
76
77         switch (ch) {
78             case 1 -> {
79                 System.out.print(s:"Name: ");
80                 String name = sc.nextLine();
81                 System.out.print(s:"Balance: ");
82                 double bal = sc.nextDouble();
83                 sc.nextLine();
84                 System.out.print(s:"Email: ");
85                 String email = sc.nextLine();
86                 System.out.print(s:"Phone: ");
87                 String phone = sc.nextLine();
88                 accounts[count] = new Account(1000 + count + 1, name, bal, email, phone);
89                 System.out.println("Account created: " + accounts[count].getAccountNumber());
90                 count++;
91             }
92             case 2 -> {
93                 System.out.print(s:"Acc No: ");
94                 int no = sc.nextInt();
95                 System.out.print(s:"Deposit: ");
96                 double amt = sc.nextDouble();
97                 Account a = find(no);
98                 if (a != null) {
99                     a.deposit(amt);
100                 } else {
101                     System.out.println(x:"Not found.");
102                 }
103             }
104             case 3 -> {
105                 System.out.print(s:"Acc No: ");
106                 int no = sc.nextInt();
107                 System.out.print(s:"Withdraw: ");
108                 double amt = sc.nextDouble();
```

```
55 public class BankingApp {
70     public static void main(String[] args) {
107         System.out.print(s: "Withdraw. ");
108         double amt = sc.nextDouble();
109         Account a = find(no);
110         if (a != null) {
111             a.withdraw(amt);
112         } else {
113             System.out.println(x: "Not found.");
114         }
115     }
116     case 4 -> {
117         System.out.print(s: "Acc No: ");
118         int no = sc.nextInt();
119         Account a = find(no);
120         if (a != null) {
121             a.show();
122         } else {
123             System.out.println(x: "Not found.");
124         }
125     }
126     case 5 -> {
127         System.out.print(s: "Acc No: ");
128         int no = sc.nextInt();
129         sc.nextLine();
130         System.out.print(s: "New Email: ");
131         String email = sc.nextLine();
132         System.out.print(s: "New Phone: ");
133         String phone = sc.nextLine();
134         Account a = find(no);
135         if (a != null) {
136             a.update(email, phone);
137         } else {
138             System.out.println(x: "Not found.");
139         }
140     }
141     case 6 -> {
142         System.out.println(x: "Exiting. Thank you!");
143         return;
```

```
55 public class BankingApp {
70     public static void main(String[] args) {
115     }
116     case 4 -> {
117         System.out.print(s:"Acc No: ");
118         int no = sc.nextInt();
119         Account a = find(no);
120         if (a != null) {
121             a.show();
122         } else {
123             System.out.println(x:"Not found.");
124         }
125     }
126     case 5 -> {
127         System.out.print(s:"Acc No: ");
128         int no = sc.nextInt();
129         sc.nextLine();
130         System.out.print(s:"New Email: ");
131         String email = sc.nextLine();
132         System.out.print(s:"New Phone: ");
133         String phone = sc.nextLine();
134         Account a = find(no);
135         if (a != null) {
136             a.update(email, phone);
137         } else {
138             System.out.println(x:"Not found.");
139         }
140     }
141     case 6 -> {
142         System.out.println(x:"Exiting. Thank you!");
143         return;
144     }
145     default ->
146         System.out.println(x:"Invalid choice!");
147 }
148 }
149 }
150 }
```

OUTPUT:

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER 1

1.Create 2.Deposit 3.Withdraw 4.View 5.Update 6.Exit
Enter choice: 1
Name: RAKESH
Balance: 1000
Email: 2401201064@gmail.com
Phone: 8610086100
Account created: 1001

1.Create 2.Deposit 3.Withdraw 4.View 5.Update 6.Exit
Enter choice: 6
Exiting. Thank you!

D:\RAKESH\VSC>JAVA ASSINGMENT 1
```