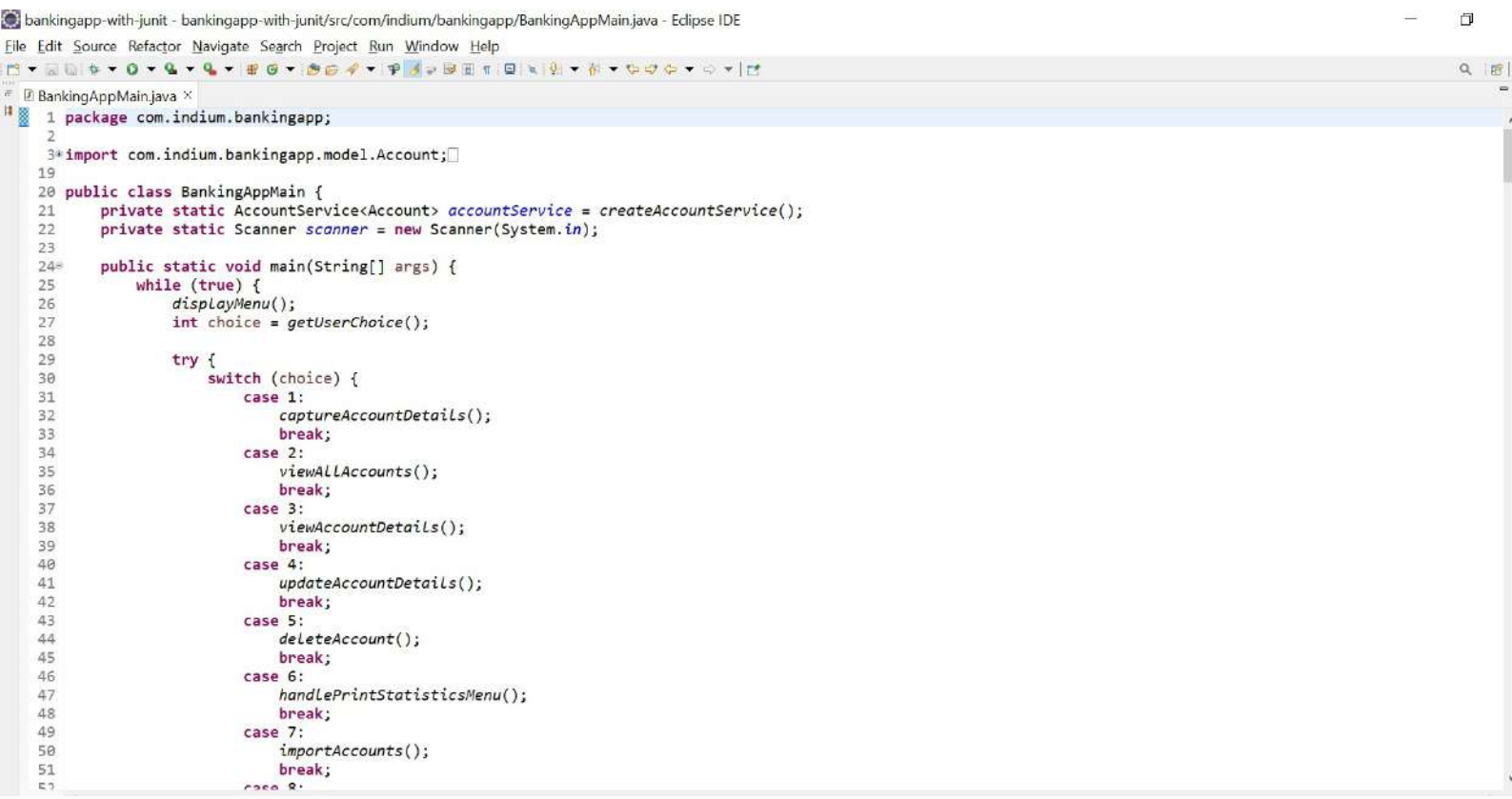
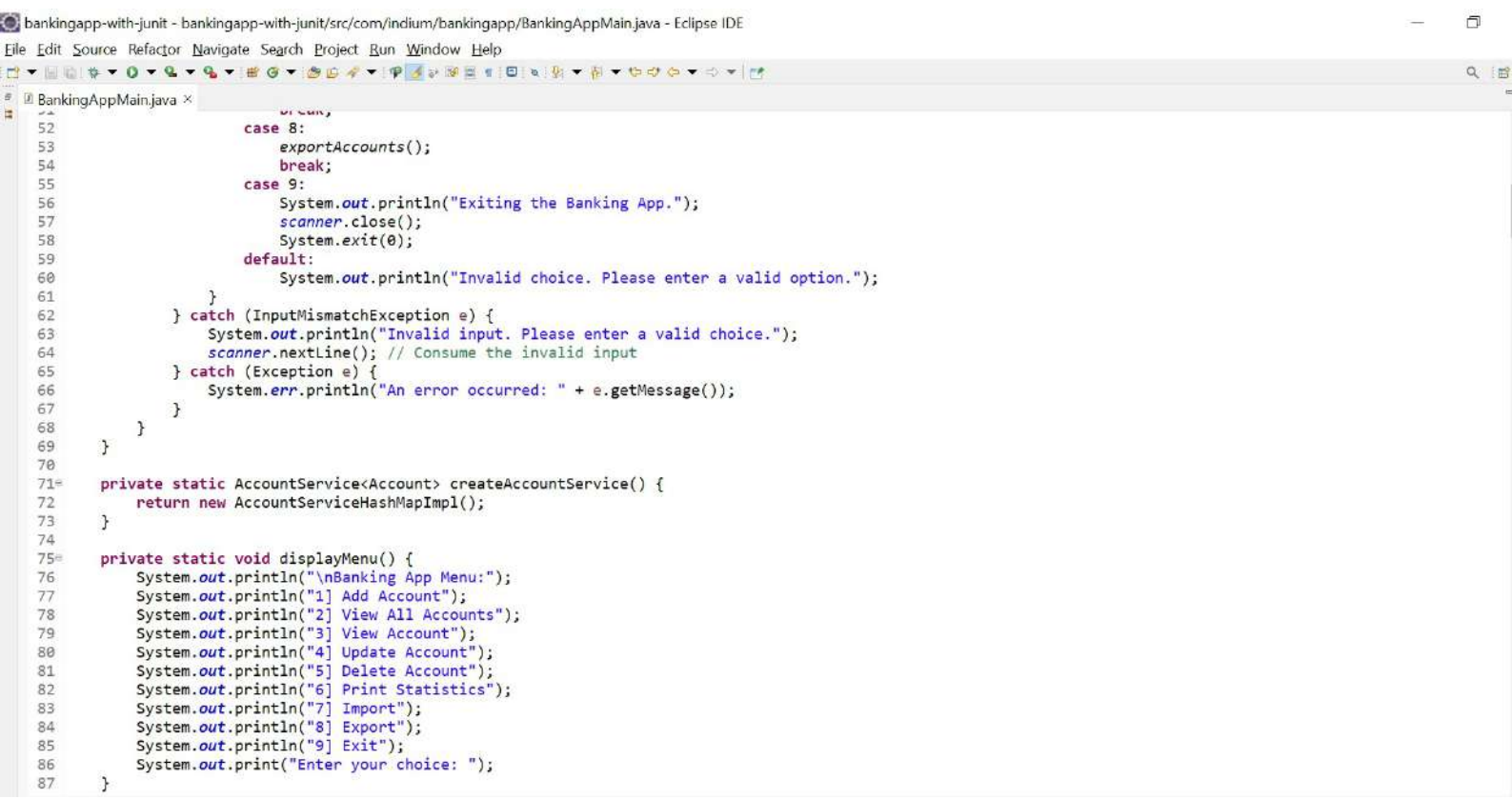

bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
1 package com.indium.bankingapp;
2
3 import com.indium.bankingapp.model.Account;
19
20 public class BankingAppMain {
21 private static AccountService<Account> accountService = createAccountService();
22 private static Scanner scanner = new Scanner(System.in);
23
24 public static void main(String[] args) {
25 while (true) {
26 displayMenu();
27 int choice = getUserChoice();
28
29 try {
30 switch (choice) {
31 case 1:
32 captureAccountDetails();
33 break;
34 case 2:
35 viewAllAccounts();
36 break;
37 case 3:
38 viewAccountDetails();
39 break;
40 case 4:
41 updateAccountDetails();
42 break;
43 case 5:
44 deleteAccount();
45 break;
46 case 6:
47 handlePrintStatisticsMenu();
48 break;
49 case 7:
50 importAccounts();
51 break;

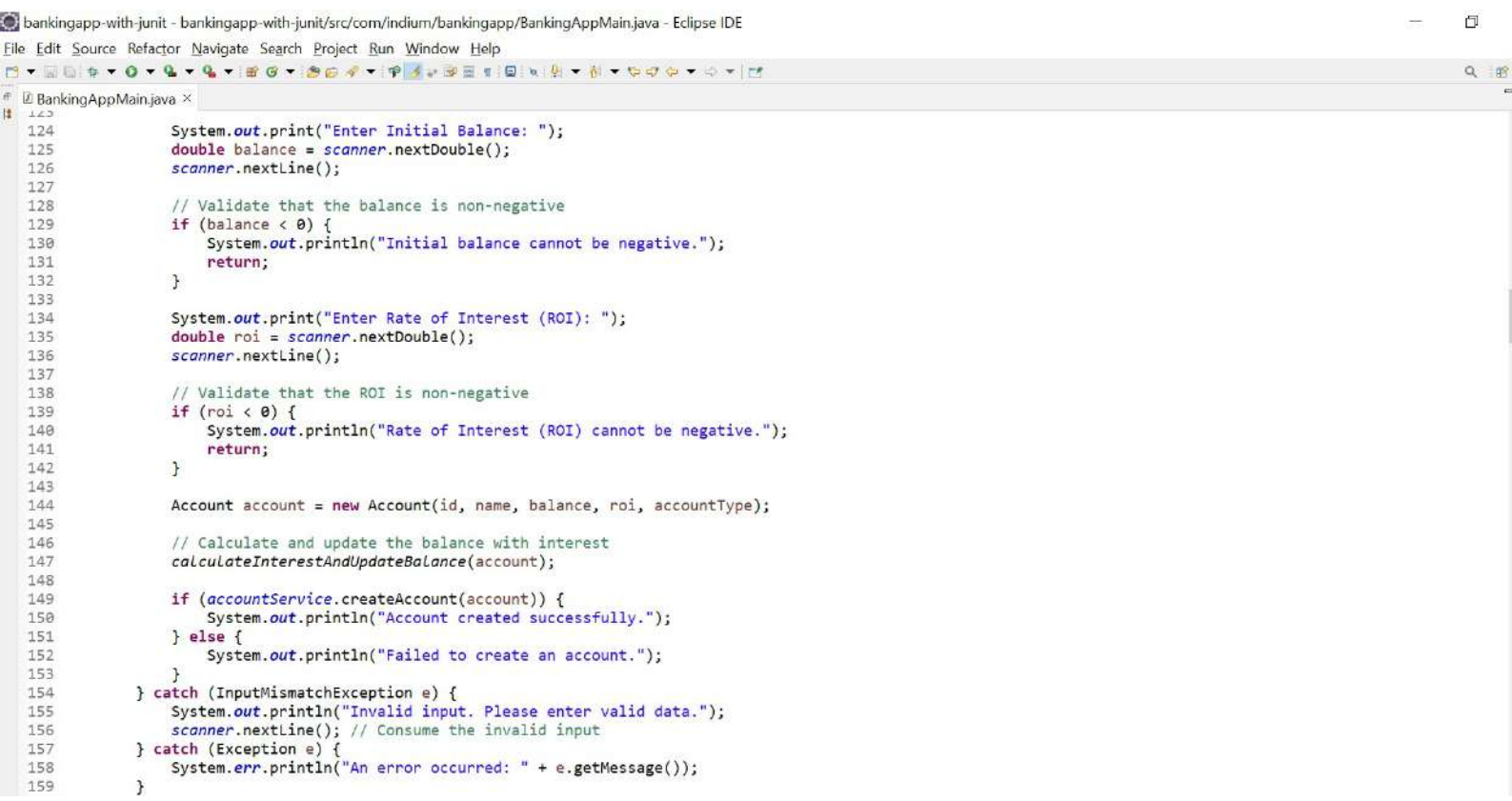

bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
52 case 8:
53 exportAccounts();
54 break;
55 case 9:
56 System.out.println("Exiting the Banking App.");
57 scanner.close();
58 System.exit(0);
59 default:
60 System.out.println("Invalid choice. Please enter a valid option.");
61 }
62 } catch (InputMismatchException e) {
63 System.out.println("Invalid input. Please enter a valid choice.");
64 scanner.nextLine(); // Consume the invalid input
65 } catch (Exception e) {
66 System.err.println("An error occurred: " + e.getMessage());
67 }
68 }
69 }
70
71 private static AccountService<Account> createAccountService() {
72 return new AccountServiceHashMapImpl();
73 }
74
75 private static void displayMenu() {
76 System.out.println("\nBanking App Menu:");
77 System.out.println("1] Add Account");
78 System.out.println("2] View All Accounts");
79 System.out.println("3] View Account");
80 System.out.println("4] Update Account");
81 System.out.println("5] Delete Account");
82 System.out.println("6] Print Statistics");
83 System.out.println("7] Import");
84 System.out.println("8] Export");
85 System.out.println("9] Exit");
86 System.out.print("Enter your choice: ");
87 }

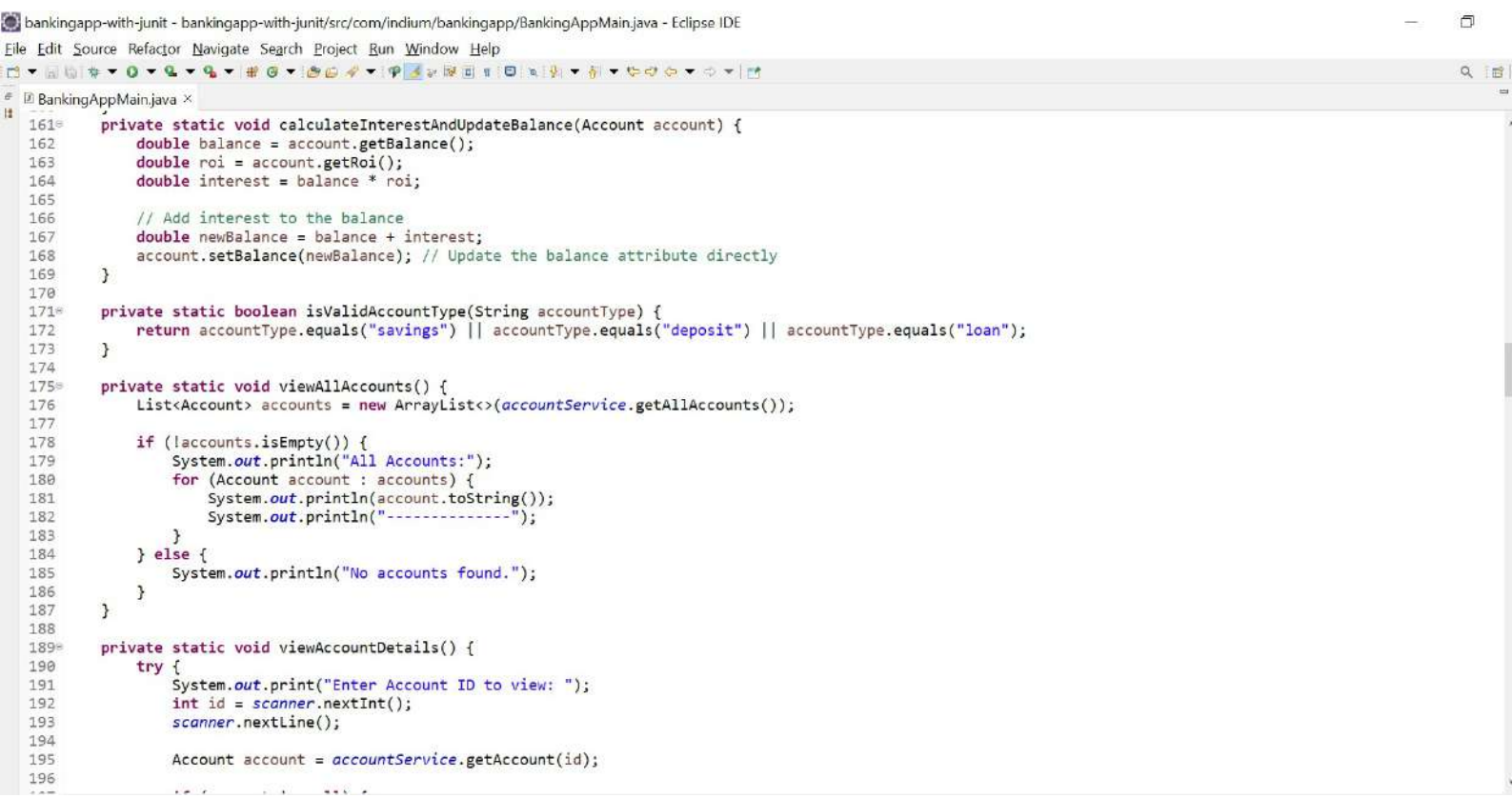
bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE

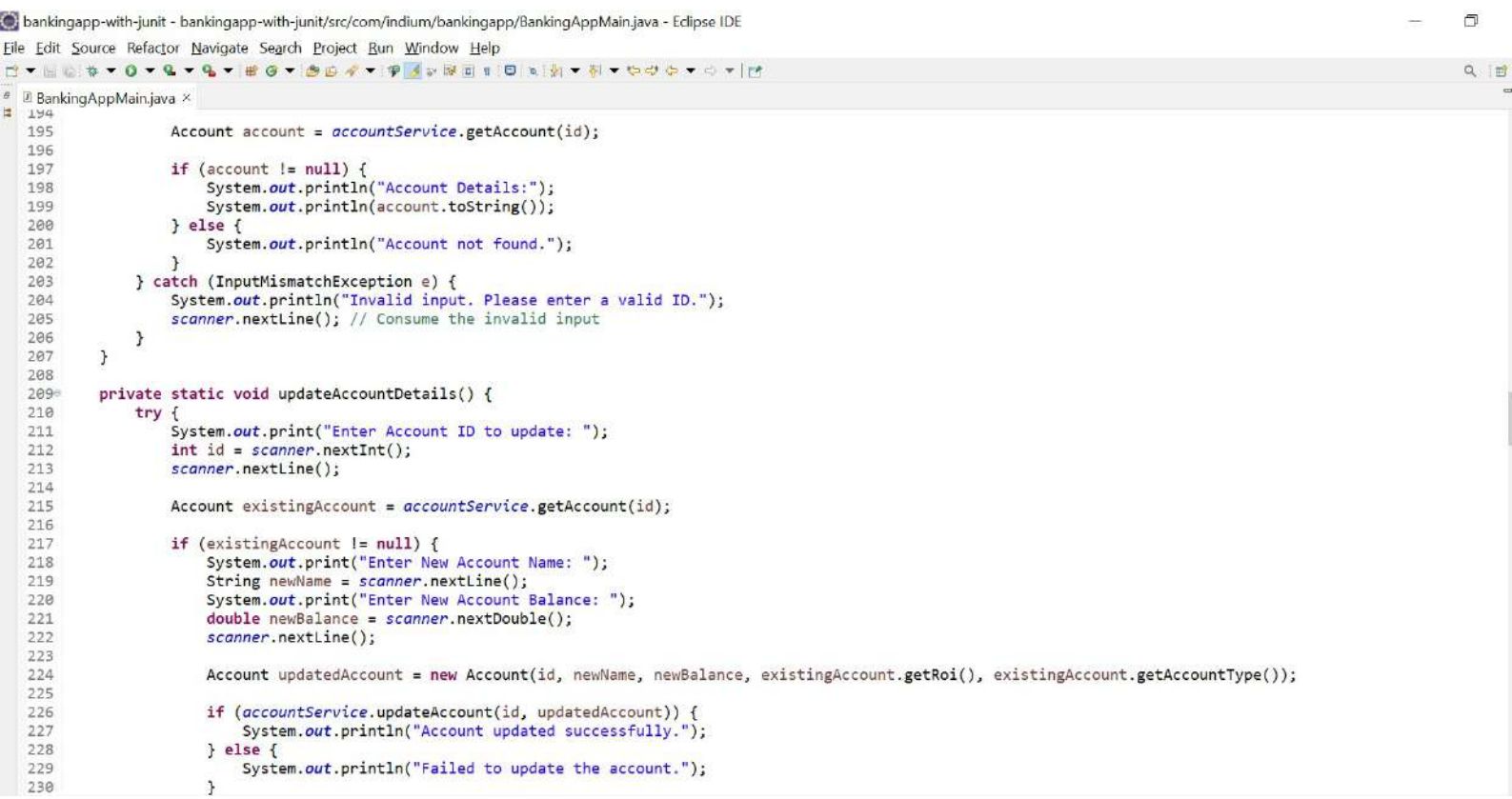
File Edit Source Refactor Navigate Search Project Run Window Help

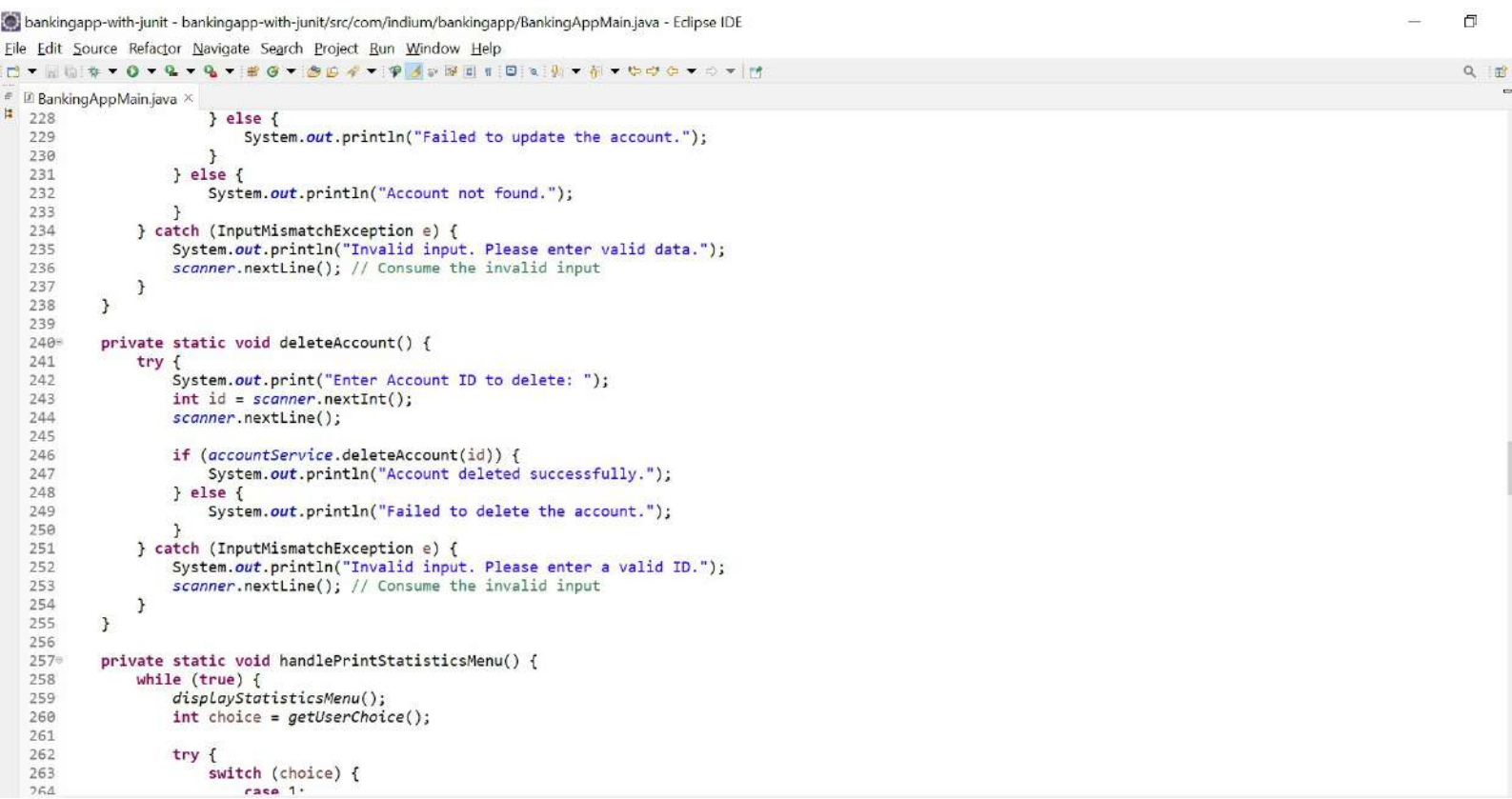
BankingAppMain.java

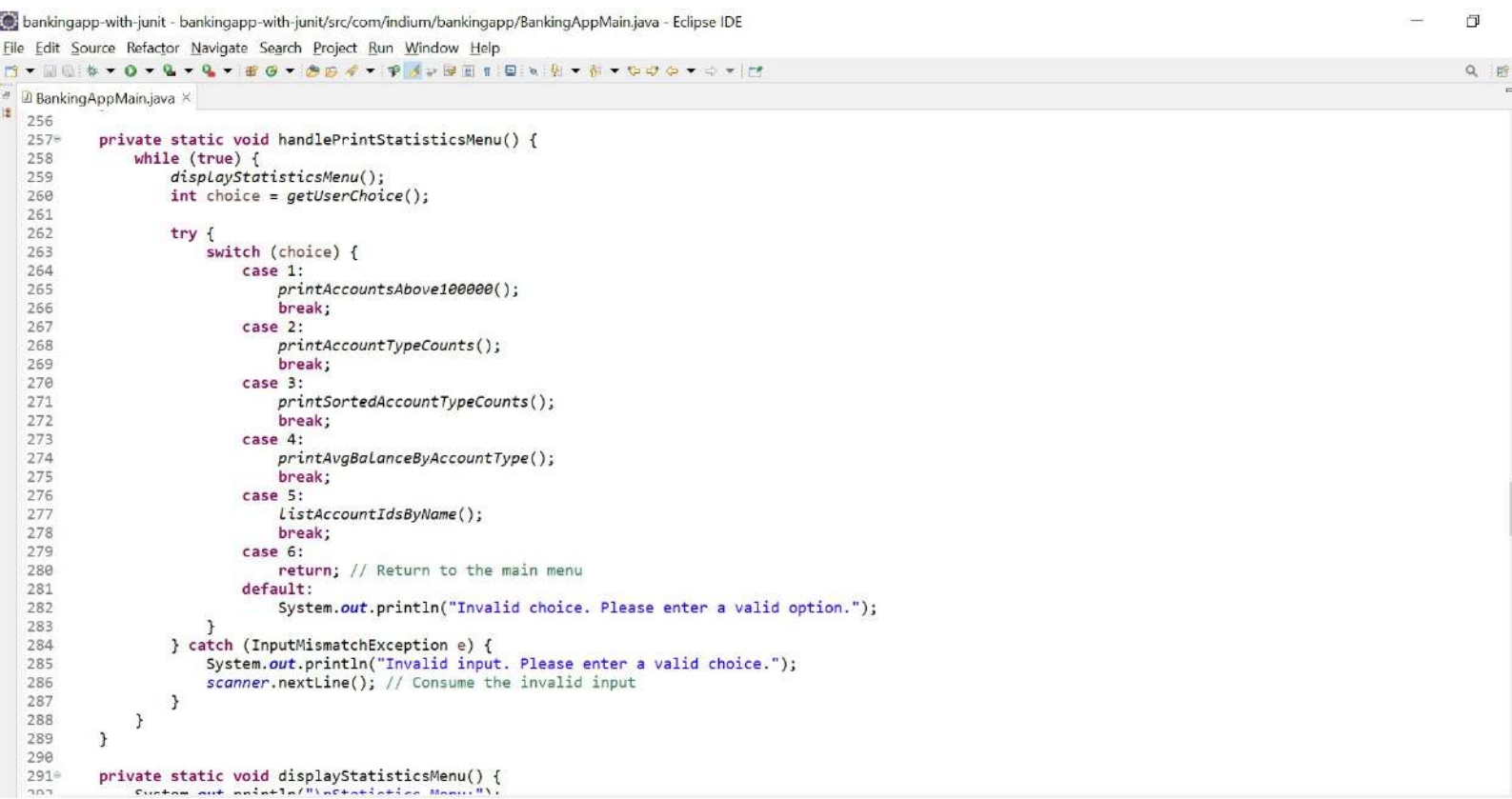
```
88
89      private static int getUserChoice() {
90          int choice = scanner.nextInt();
91          scanner.nextLine();
92          return choice;
93      }
94
95      private static void captureAccountDetails() {
96          try {
97              System.out.print("Enter Account Type (savings/deposit/loan): ");
98              String accountType = scanner.nextLine().toLowerCase();
99
100             if (!isValidAccountType(accountType)) {
101                 System.out.println("Invalid account type. Please enter 'savings', 'deposit', or 'loan'.");
102                 return;
103             }
104
105             System.out.print("Enter Account ID: ");
106             int id = scanner.nextInt();
107             scanner.nextLine();
108
109             // Validate that ID is positive
110             if (id <= 0) {
111                 System.out.println("Invalid account ID. Please enter a positive integer.");
112                 return;
113             }
114
115             System.out.print("Enter Account Name: ");
116             String name = scanner.nextLine();
117
118             // Validate that the name is not empty
119             if (name.isEmpty()) {
120                 System.out.println("Account name cannot be empty.");
121                 return;
122             }
123
124             System.out.print("Enter Initial Balance: ");
```


bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
System.out.print("Enter Initial Balance: ");
double balance = scanner.nextDouble();
scanner.nextLine();
// Validate that the balance is non-negative
if (balance < 0) {
System.out.println("Initial balance cannot be negative.");
return;
}
System.out.print("Enter Rate of Interest (ROI): ");
double roi = scanner.nextDouble();
scanner.nextLine();
// Validate that the ROI is non-negative
if (roi < 0) {
System.out.println("Rate of Interest (ROI) cannot be negative.");
return;
}
Account account = new Account(id, name, balance, roi, accountType);
// Calculate and update the balance with interest
calculateInterestAndUpdateBalance(account);
if (accountService.createAccount(account)) {
System.out.println("Account created successfully.");
} else {
System.out.println("Failed to create an account.");
}
} catch (InputMismatchException e) {
System.out.println("Invalid input. Please enter valid data.");
scanner.nextLine(); // Consume the invalid input
} catch (Exception e) {
System.err.println("An error occurred: " + e.getMessage());
}

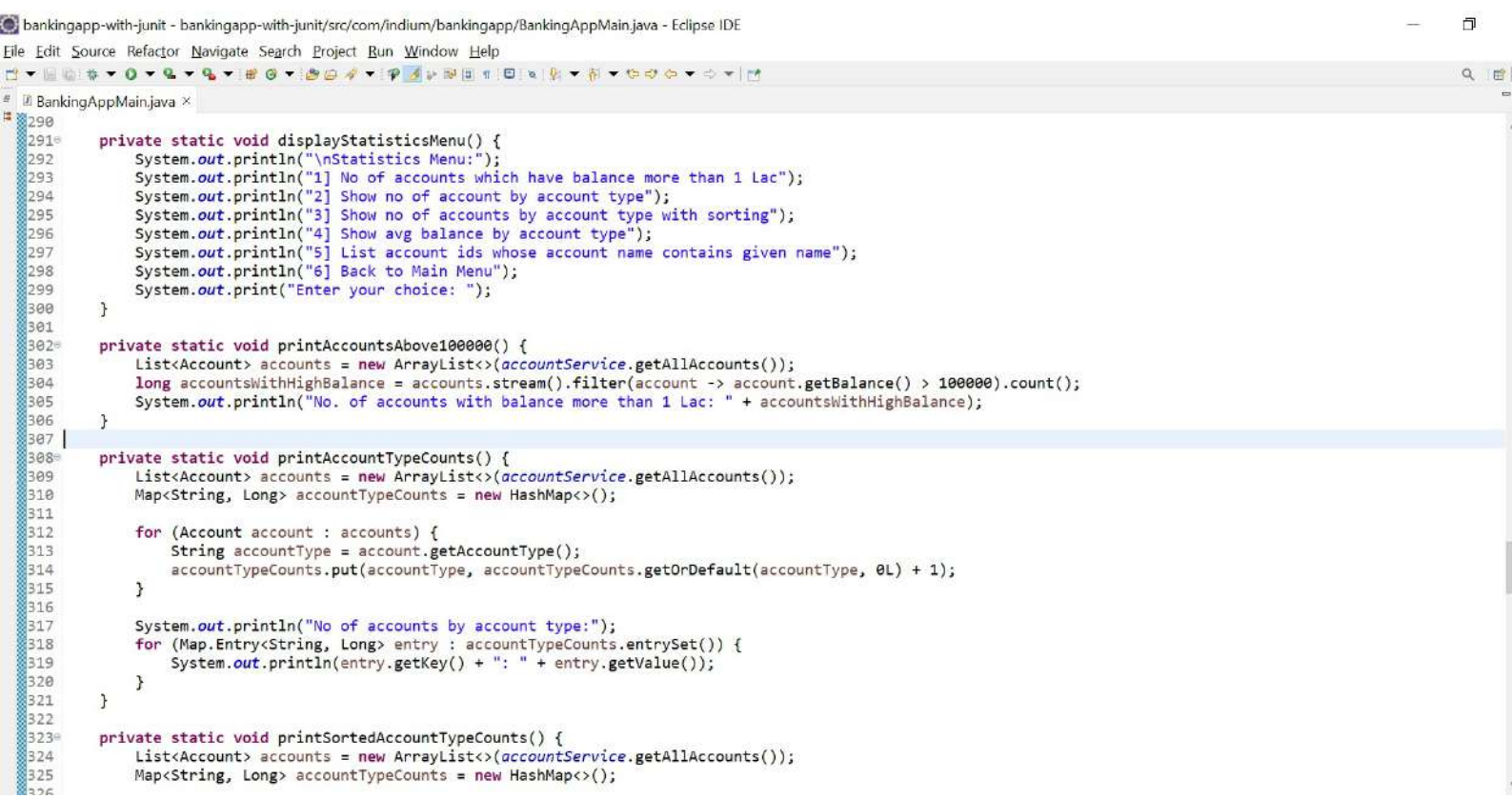

bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
161 private static void calculateInterestAndUpdateBalance(Account account) {
162 double balance = account.getBalance();
163 double roi = account.getRoi();
164 double interest = balance * roi;
165
166 // Add interest to the balance
167 double newBalance = balance + interest;
168 account.setBalance(newBalance); // Update the balance attribute directly
169 }
170
171 private static boolean isValidAccountType(String accountType) {
172 return accountType.equals("savings") || accountType.equals("deposit") || accountType.equals("loan");
173 }
174
175 private static void viewAllAccounts() {
176 List<Account> accounts = new ArrayList<>(accountService.getAllAccounts());
177
178 if (!accounts.isEmpty()) {
179 System.out.println("All Accounts:");
180 for (Account account : accounts) {
181 System.out.println(account.toString());
182 System.out.println("--------------");
183 }
184 } else {
185 System.out.println("No accounts found.");
186 }
187 }
188
189 private static void viewAccountDetails() {
190 try {
191 System.out.print("Enter Account ID to view: ");
192 int id = scanner.nextInt();
193 scanner.nextLine();
194
195 Account account = accountService.getAccount(id);
196

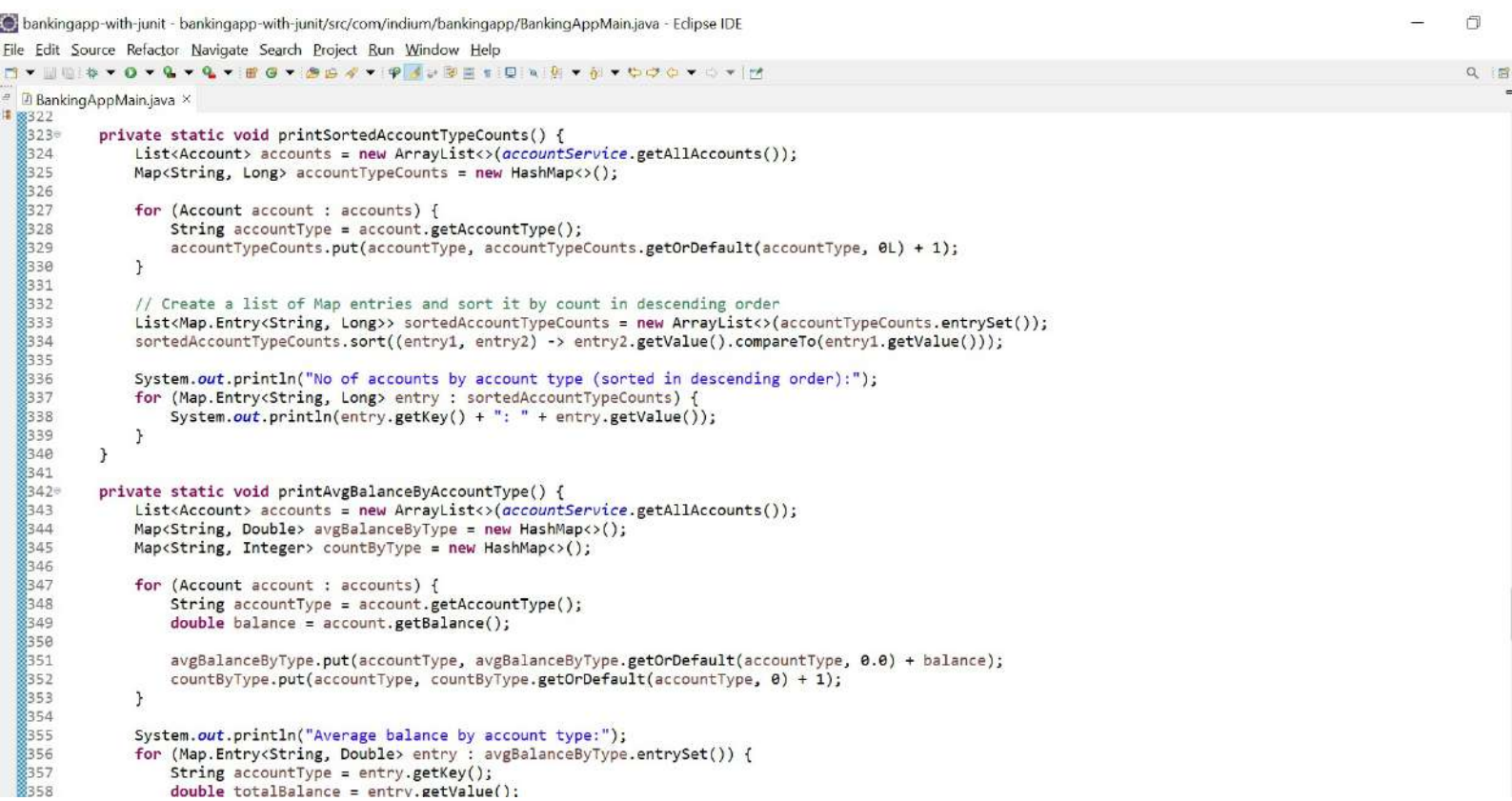

bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
Account account = accountService.getAccount(id);
if (account != null) {
System.out.println("Account Details:");
System.out.println(account.toString());
} else {
System.out.println("Account not found.");
}
} catch (InputMismatchException e) {
System.out.println("Invalid input. Please enter a valid ID.");
scanner.nextLine(); // Consume the invalid input
}
}
private static void updateAccountDetails() {
try {
System.out.print("Enter Account ID to update: ");
int id = scanner.nextInt();
scanner.nextLine();
Account existingAccount = accountService.getAccount(id);
if (existingAccount != null) {
System.out.print("Enter New Account Name: ");
String newName = scanner.nextLine();
System.out.print("Enter New Account Balance: ");
double newBalance = scanner.nextDouble();
scanner.nextLine();
Account updatedAccount = new Account(id, newName, newBalance, existingAccount.getRoi(), existingAccount.getAccountType());
if (accountService.updateAccount(id, updatedAccount)) {
System.out.println("Account updated successfully.");
} else {
System.out.println("Failed to update the account.");
}

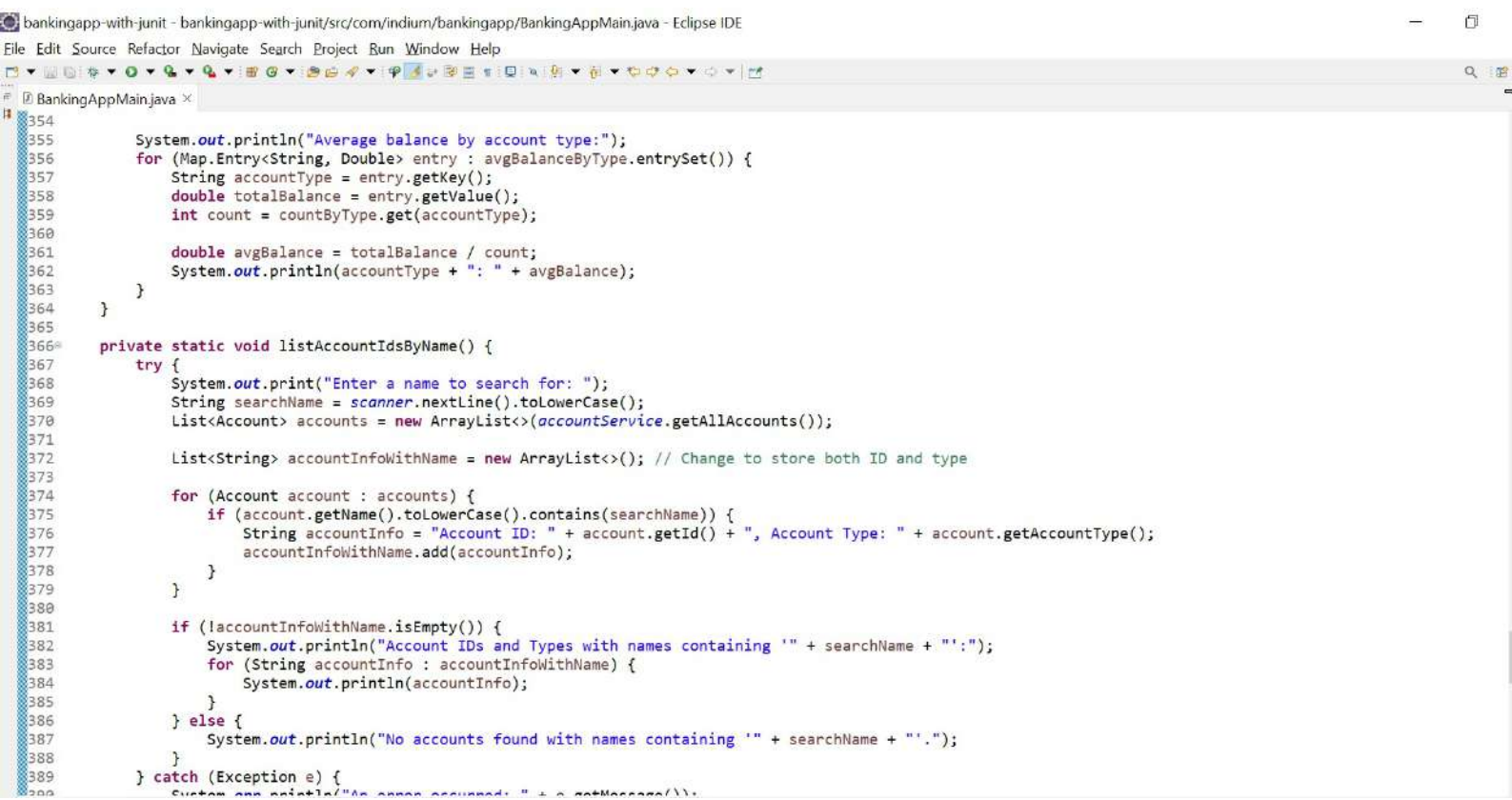

bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
228 } else {
229 System.out.println("Failed to update the account.");
230 }
231 } else {
232 System.out.println("Account not found.");
233 }
234 } catch (InputMismatchException e) {
235 System.out.println("Invalid input. Please enter valid data.");
236 scanner.nextLine(); // Consume the invalid input
237 }
238 }
239
240 private static void deleteAccount() {
241 try {
242 System.out.print("Enter Account ID to delete: ");
243 int id = scanner.nextInt();
244 scanner.nextLine();
245
246 if (accountService.deleteAccount(id)) {
247 System.out.println("Account deleted successfully.");
248 } else {
249 System.out.println("Failed to delete the account.");
250 }
251 } catch (InputMismatchException e) {
252 System.out.println("Invalid input. Please enter a valid ID.");
253 scanner.nextLine(); // Consume the invalid input
254 }
255 }
256
257 private static void handlePrintStatisticsMenu() {
258 while (true) {
259 displayStatisticsMenu();
260 int choice = getUserChoice();
261
262 try {
263 switch (choice) {
264 case 1:



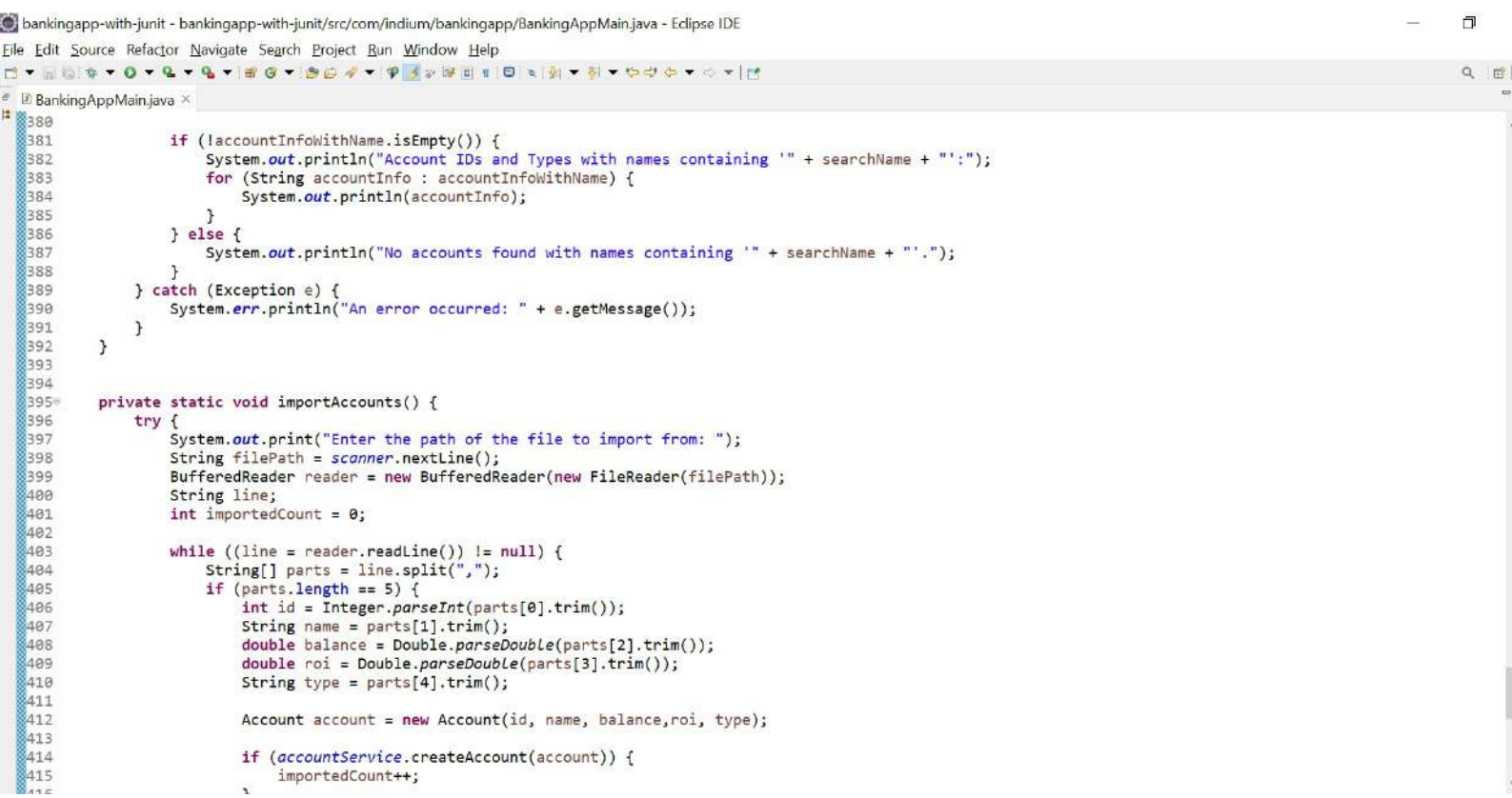

```java
256
257    private static void handlePrintStatisticsMenu() {
258        while (true) {
259            displayStatisticsMenu();
260            int choice = getUserChoice();
261
262            try {
263                switch (choice) {
264                    case 1:
265                        printAccountsAbove100000();
266                        break;
267                    case 2:
268                        printAccountTypeCounts();
269                        break;
270                    case 3:
271                        printSortedAccountTypeCounts();
272                        break;
273                    case 4:
274                        printAvgBalanceByAccountType();
275                        break;
276                    case 5:
277                        listAccountIdsByName();
278                        break;
279                    case 6:
280                        return; // Return to the main menu
281                    default:
282                        System.out.println("Invalid choice. Please enter a valid option.");
283                }
284            } catch (InputMismatchException e) {
285                System.out.println("Invalid input. Please enter a valid choice.");
286                scanner.nextLine(); // Consume the invalid input
287            }
288        }
289    }
290
291    private static void displayStatisticsMenu() {
```

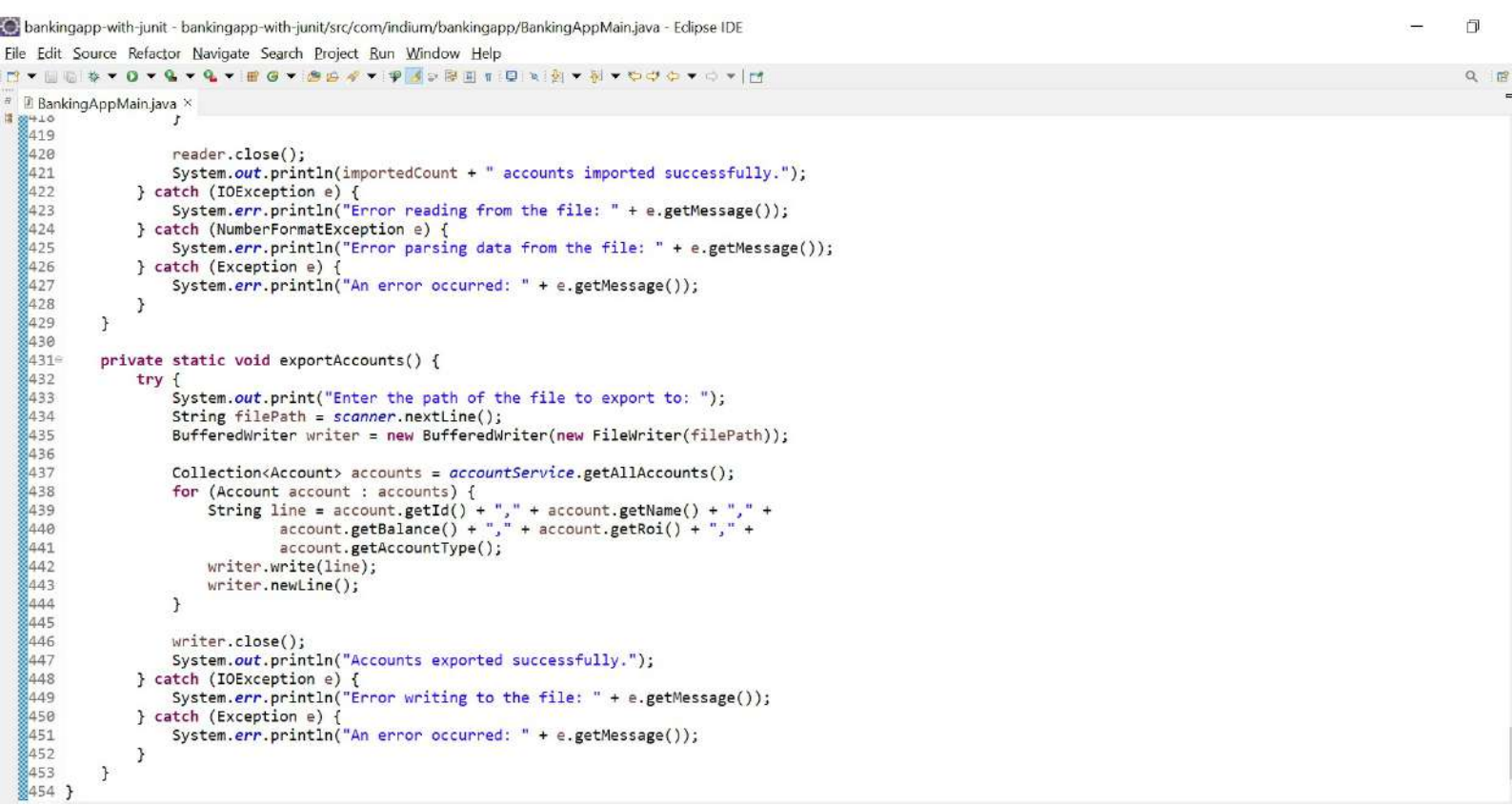

bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
290
291 private static void displayStatisticsMenu() {
292 System.out.println("\nStatistics Menu:");
293 System.out.println("1] No of accounts which have balance more than 1 Lac");
294 System.out.println("2] Show no of account by account type");
295 System.out.println("3] Show no of accounts by account type with sorting");
296 System.out.println("4] Show avg balance by account type");
297 System.out.println("5] List account ids whose account name contains given name");
298 System.out.println("6] Back to Main Menu");
299 System.out.print("Enter your choice: ");
300 }
301
302 private static void printAccountsAbove100000() {
303 List<Account> accounts = new ArrayList<>(accountService.getAllAccounts());
304 long accountsWithHighBalance = accounts.stream().filter(account -> account.getBalance() > 100000).count();
305 System.out.println("No. of accounts with balance more than 1 Lac: " + accountsWithHighBalance);
306 }
307
308 private static void printAccountTypeCounts() {
309 List<Account> accounts = new ArrayList<>(accountService.getAllAccounts());
310 Map<String, Long> accountTypeCounts = new HashMap<>();
311
312 for (Account account : accounts) {
313 String accountType = account.getAccountType();
314 accountTypeCounts.put(accountType, accountTypeCounts.getOrDefault(accountType, 0L) + 1);
315 }
316
317 System.out.println("No of accounts by account type:");
318 for (Map.Entry<String, Long> entry : accountTypeCounts.entrySet()) {
319 System.out.println(entry.getKey() + ": " + entry.getValue());
320 }
321 }
322
323 private static void printSortedAccountTypeCounts() {
324 List<Account> accounts = new ArrayList<>(accountService.getAllAccounts());
325 Map<String, Long> accountTypeCounts = new HashMap<>();


bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
322
323 private static void printSortedAccountTypeCounts() {
324 List<Account> accounts = new ArrayList<>(accountService.getAllAccounts());
325 Map<String, Long> accountTypeCounts = new HashMap<>();
326
327 for (Account account : accounts) {
328 String accountType = account.getAccountType();
329 accountTypeCounts.put(accountType, accountTypeCounts.getOrDefault(accountType, 0L) + 1);
330 }
331
332 // Create a list of Map entries and sort it by count in descending order
333 List<Map.Entry<String, Long>> sortedAccountTypeCounts = new ArrayList<>(accountTypeCounts.entrySet());
334 sortedAccountTypeCounts.sort((entry1, entry2) -> entry2.getValue().compareTo(entry1.getValue()));
335
336 System.out.println("No of accounts by account type (sorted in descending order):");
337 for (Map.Entry<String, Long> entry : sortedAccountTypeCounts) {
338 System.out.println(entry.getKey() + ": " + entry.getValue());
339 }
340 }
341
342 private static void printAvgBalanceByAccountType() {
343 List<Account> accounts = new ArrayList<>(accountService.getAllAccounts());
344 Map<String, Double> avgBalanceByType = new HashMap<>();
345 Map<String, Integer> countByType = new HashMap<>();
346
347 for (Account account : accounts) {
348 String accountType = account.getAccountType();
349 double balance = account.getBalance();
350
351 avgBalanceByType.put(accountType, avgBalanceByType.getOrDefault(accountType, 0.0) + balance);
352 countByType.put(accountType, countByType.getOrDefault(accountType, 0) + 1);
353 }
354
355 System.out.println("Average balance by account type:");
356 for (Map.Entry<String, Double> entry : avgBalanceByType.entrySet()) {
357 String accountType = entry.getKey();
358 double totalBalance = entry.getValue();




```
354
355         System.out.println("Average balance by account type:");
356         for (Map.Entry<String, Double> entry : avgBalanceByType.entrySet()) {
357             String accountType = entry.getKey();
358             double totalBalance = entry.getValue();
359             int count = countByType.get(accountType);
360
361             double avgBalance = totalBalance / count;
362             System.out.println(accountType + ": " + avgBalance);
363         }
364     }
365
366     private static void listAccountIdsByName() {
367         try {
368             System.out.print("Enter a name to search for: ");
369             String searchName = scanner.nextLine().toLowerCase();
370             List<Account> accounts = new ArrayList<>(accountService.getAllAccounts());
371
372             List<String> accountInfoWithName = new ArrayList<>(); // Change to store both ID and type
373
374             for (Account account : accounts) {
375                 if (account.getName().toLowerCase().contains(searchName)) {
376                     String accountInfo = "Account ID: " + account.getId() + ", Account Type: " + account.getAccountType();
377                     accountInfoWithName.add(accountInfo);
378                 }
379             }
380
381             if (!accountInfoWithName.isEmpty()) {
382                 System.out.println("Account IDs and Types with names containing '" + searchName + "':");
383                 for (String accountInfo : accountInfoWithName) {
384                     System.out.println(accountInfo);
385                 }
386             } else {
387                 System.out.println("No accounts found with names containing '" + searchName + "'.");
388             }
389         } catch (Exception e) {
```


bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
380
381 if (!accountInfoWithName.isEmpty()) {
382 System.out.println("Account IDs and Types with names containing '" + searchName + "':");
383 for (String accountInfo : accountInfoWithName) {
384 System.out.println(accountInfo);
385 }
386 } else {
387 System.out.println("No accounts found with names containing '" + searchName + "'.");
388 }
389 } catch (Exception e) {
390 System.err.println("An error occurred: " + e.getMessage());
391 }
392 }
393
394
395 private static void importAccounts() {
396 try {
397 System.out.print("Enter the path of the file to import from: ");
398 String filePath = scanner.nextLine();
399 BufferedReader reader = new BufferedReader(new FileReader(filePath));
400 String line;
401 int importedCount = 0;
402
403 while ((line = reader.readLine()) != null) {
404 String[] parts = line.split(",");
405 if (parts.length == 5) {
406 int id = Integer.parseInt(parts[0].trim());
407 String name = parts[1].trim();
408 double balance = Double.parseDouble(parts[2].trim());
409 double roi = Double.parseDouble(parts[3].trim());
410 String type = parts[4].trim();
411
412 Account account = new Account(id, name, balance,roi, type);
413
414 if (accountService.createAccount(account)) {
415 importedCount++;


bankingapp-with-junit - bankingapp-with-junit/src/com/indium/bankingapp/BankingAppMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankingAppMain.java
}
reader.close();
System.out.println(importedCount + " accounts imported successfully.");
} catch (IOException e) {
System.err.println("Error reading from the file: " + e.getMessage());
} catch (NumberFormatException e) {
System.err.println("Error parsing data from the file: " + e.getMessage());
} catch (Exception e) {
System.err.println("An error occurred: " + e.getMessage());
}
}
private static void exportAccounts() {
try {
System.out.print("Enter the path of the file to export to: ");
String filePath = scanner.nextLine();
BufferedWriter writer = new BufferedWriter(new FileWriter(filePath));
Collection<Account> accounts = accountService.getAllAccounts();
for (Account account : accounts) {
String line = account.getId() + "," + account.getName() + "," +
account.getBalance() + "," + account.getRoi() + "," +
account.getAccountType();
writer.write(line);
writer.newLine();
}
writer.close();
System.out.println("Accounts exported successfully.");
} catch (IOException e) {
System.err.println("Error writing to the file: " + e.getMessage());
} catch (Exception e) {
System.err.println("An error occurred: " + e.getMessage());
}
}
}