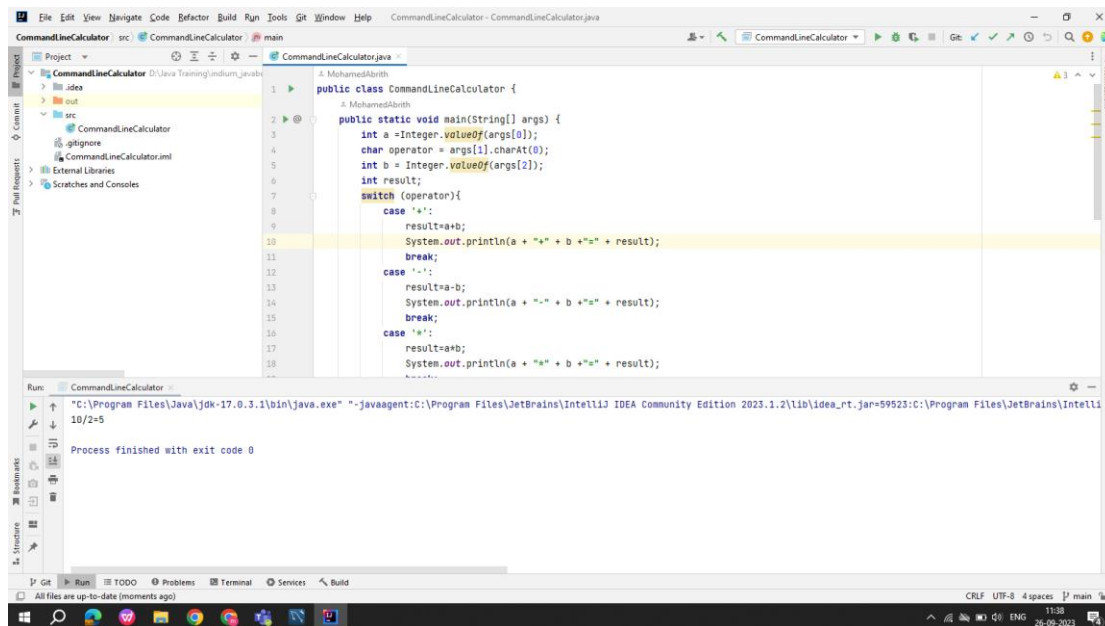
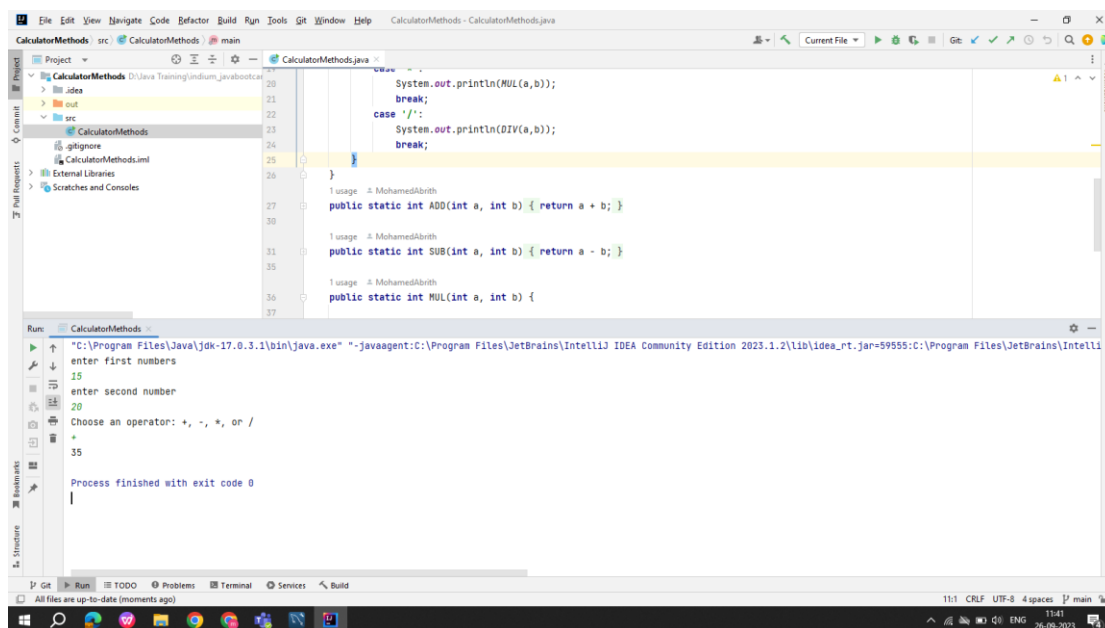


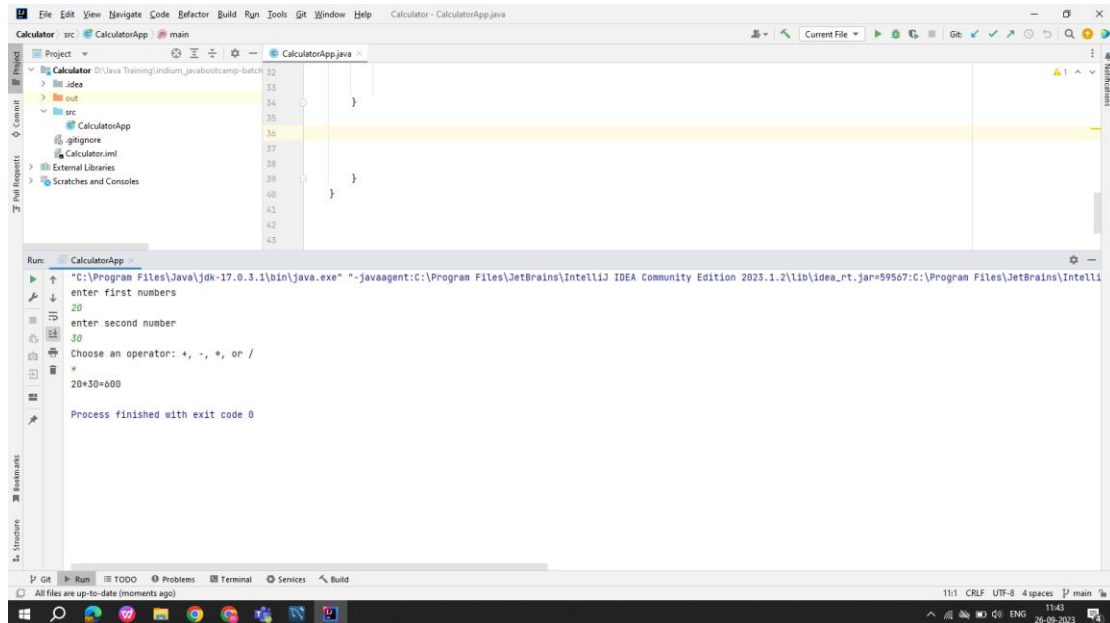
CommandLineCalculator



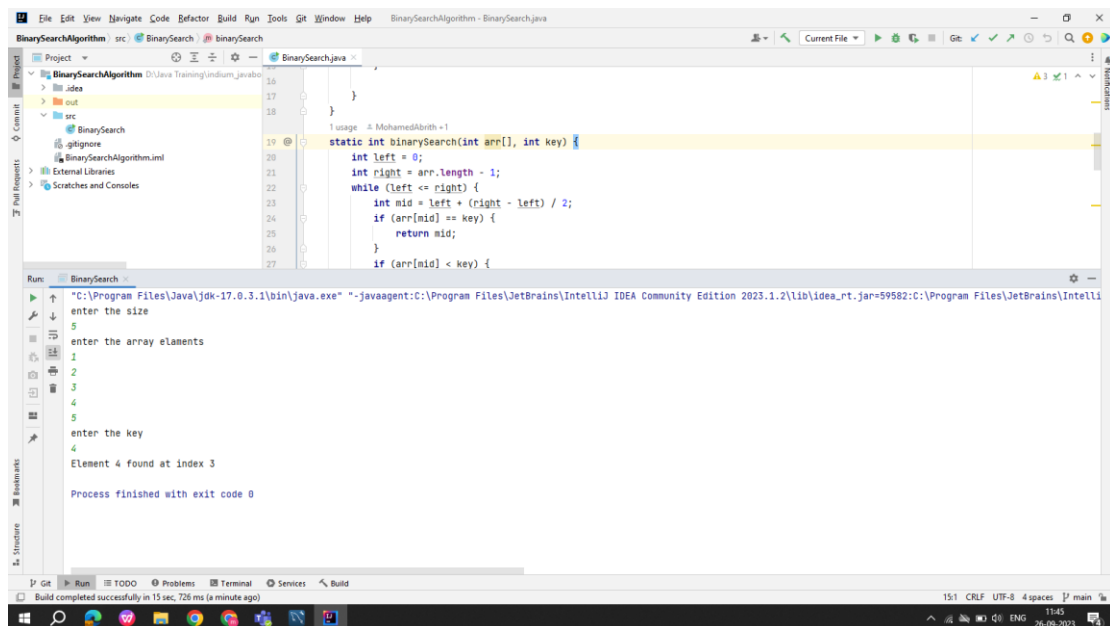
CalculatorMethods



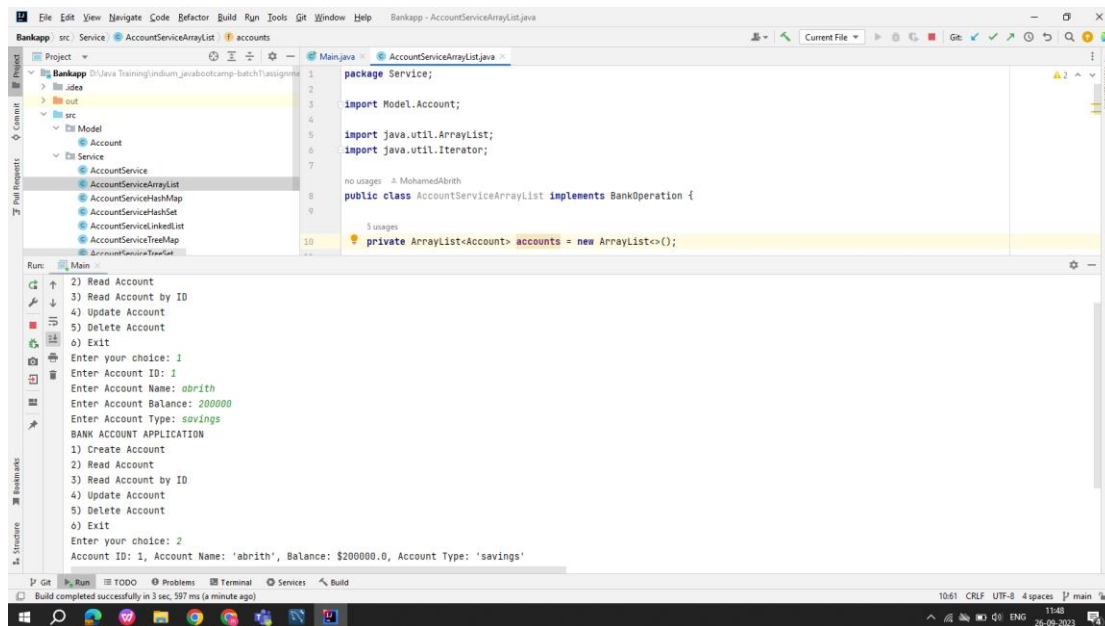
Calculator



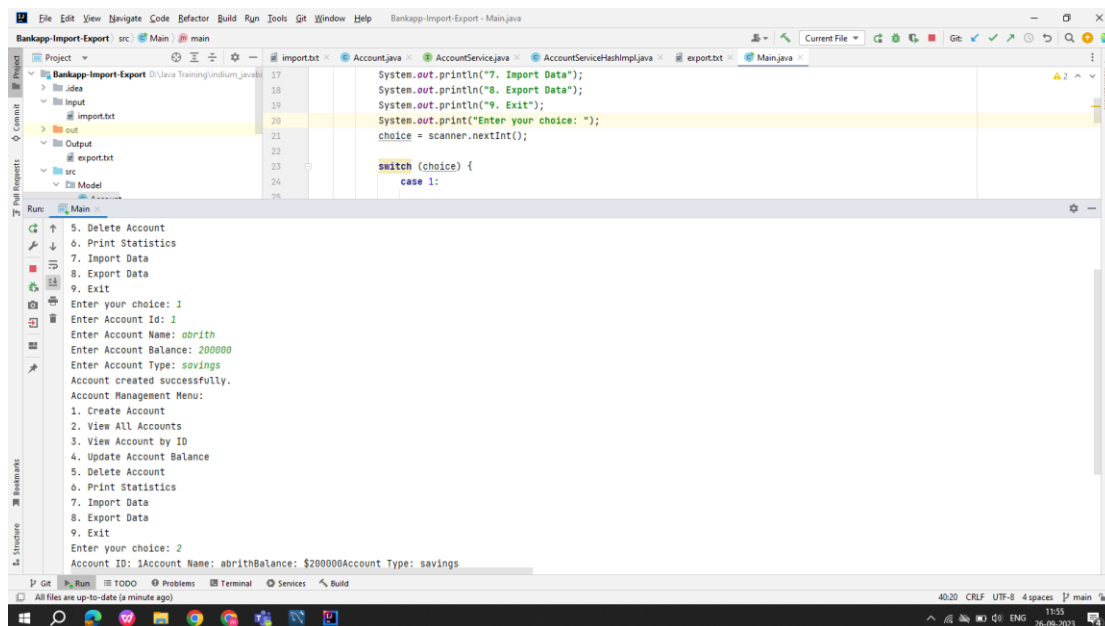
BinarySearchAlgorithm

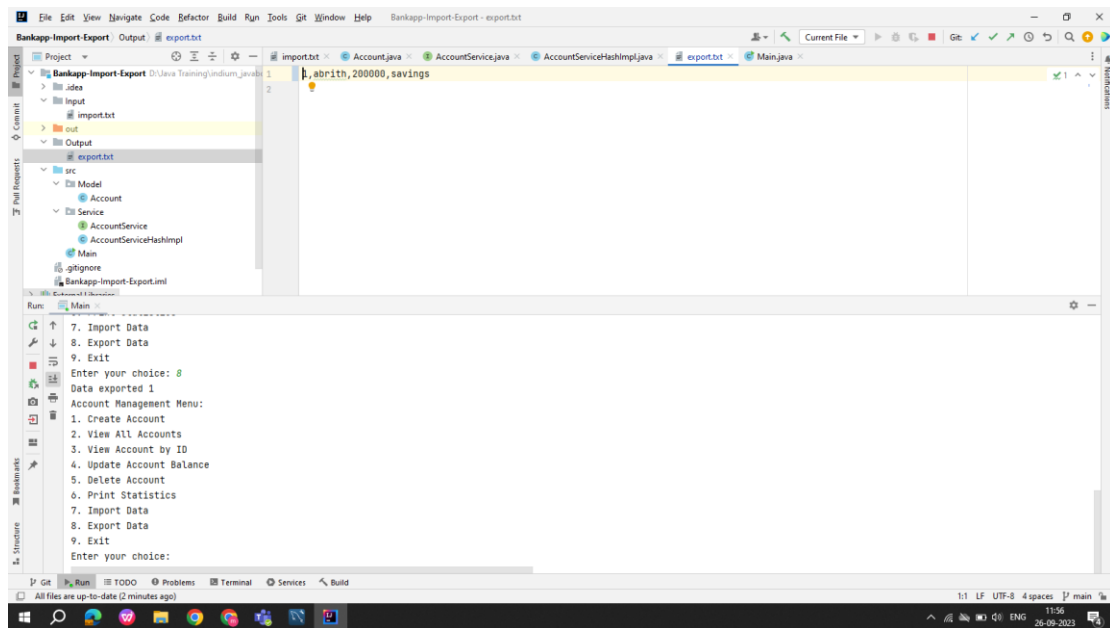


Bankapp

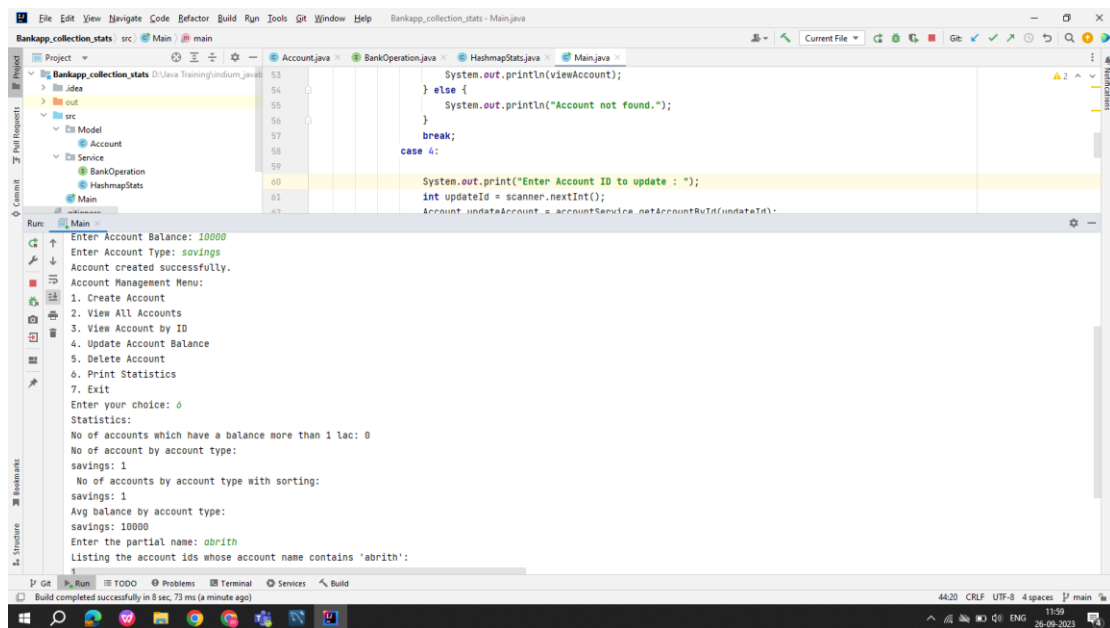


Bankapp-Import-Export





Bankapp_collection_stats



The screenshot shows an IDE with the following components:

- Menu Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Git, Window, Help.
- Project Explorer (Left):** Shows the project structure: Bankapp-with-threads > src > Main > printStatistics. The 'import.txt' file is highlighted.
- Main Window:** Displays the 'Main.java' file. The code includes a 'Runnable' interface and a 'Runnable' implementation. The 'run()' method contains a loop that prints a menu and handles user input. The 'importThread()' method is also visible.
- Run Console (Bottom):** Shows the execution output. It starts with 'Account Management Menu:' and a list of options. The user enters '7', and the output shows 'Thread[pool-1-thread-1,main] is importing data' followed by '1,abrith,200000,deposit' and 'Data Imported'.
- Structure Panel (Bottom Left):** Shows the project structure, including 'Main.java' and 'import.txt'.
- Status Bar (Bottom):** Shows the current file is up-to-date (3 minutes ago).

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `src`, `com.indium.bankapp`, `Model`, `Account`, `Service`, `AccountService`, `AccountServiceHashMap`, `Main`, `gignore`, and `Bankapp-with-stream.iml`.
- Code Editor:**
 - `AccountService.java`: Contains the `deleteAccount` method implementation:


```
1 usage: MohamedAbriht
@Override
public void deleteAccount(int id) { accountMap.remove(id); }

1 usage: MohamedAbriht
@Override
public long countAccountsAboveOneLac() {
    return accountMap.values().Collection<Account>()
        .stream().Stream<Account>()
        .filter(account -> account.getBalance() > 100000)
        .count();
}
```
 - `Main.java`: Contains the `main` method with user input and menu options. The current state shows:


```
Enter Account Name: abriht
Enter Account Balance: 20000
Enter Account Type: savings
Account created successfully.
Account Management Menu:
1. Create Account
2. View All Accounts
3. View Account by ID
4. Update Account Balance
5. Delete Account
6. Print Statistics
7. Import Data
8. Export Data
9. Exit
Enter your choice: 2
Account ID: 1 Account Name: abrihtBalance: $20000Account Type: savings
Account Management Menu:
```
- Run Console:** Shows the output of the program, including the menu and user interactions.

Bankapp-with-JDBC

