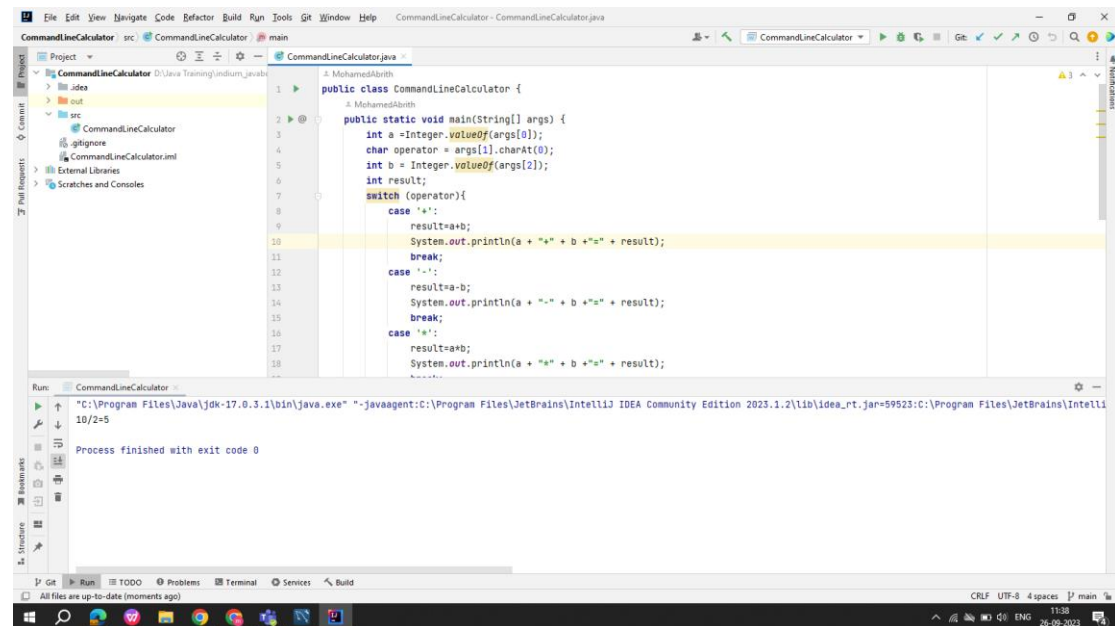
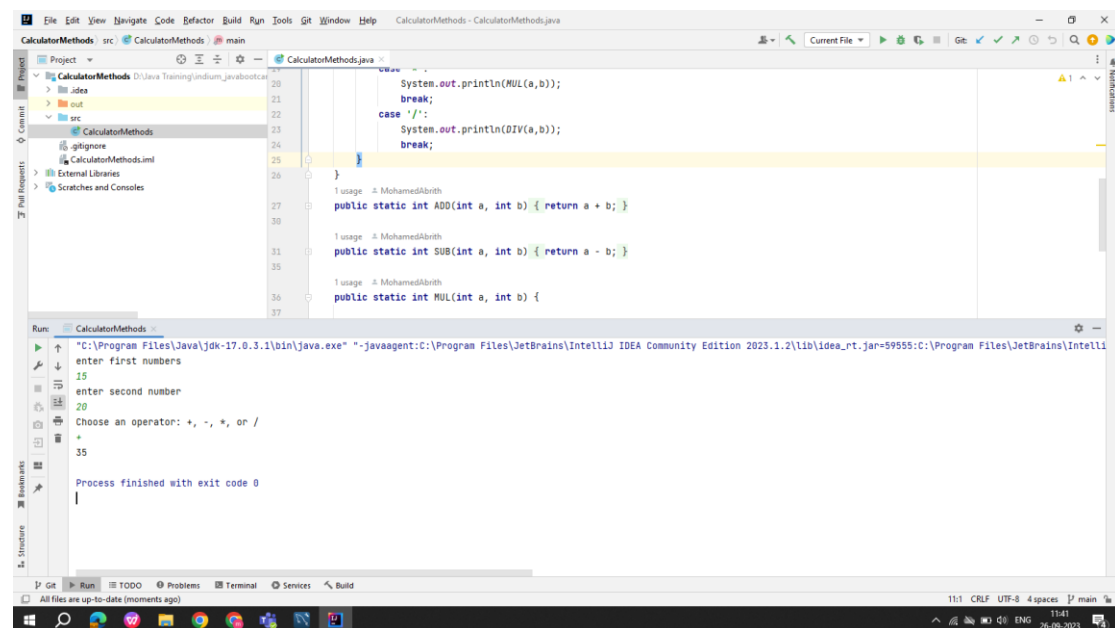


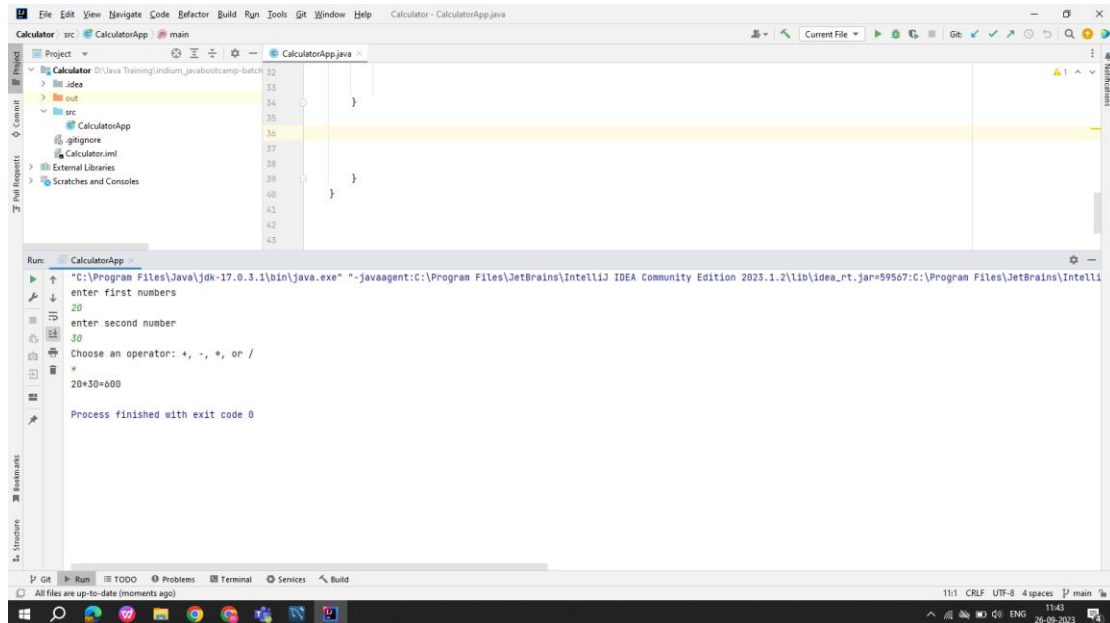
# CommandLineCalculator



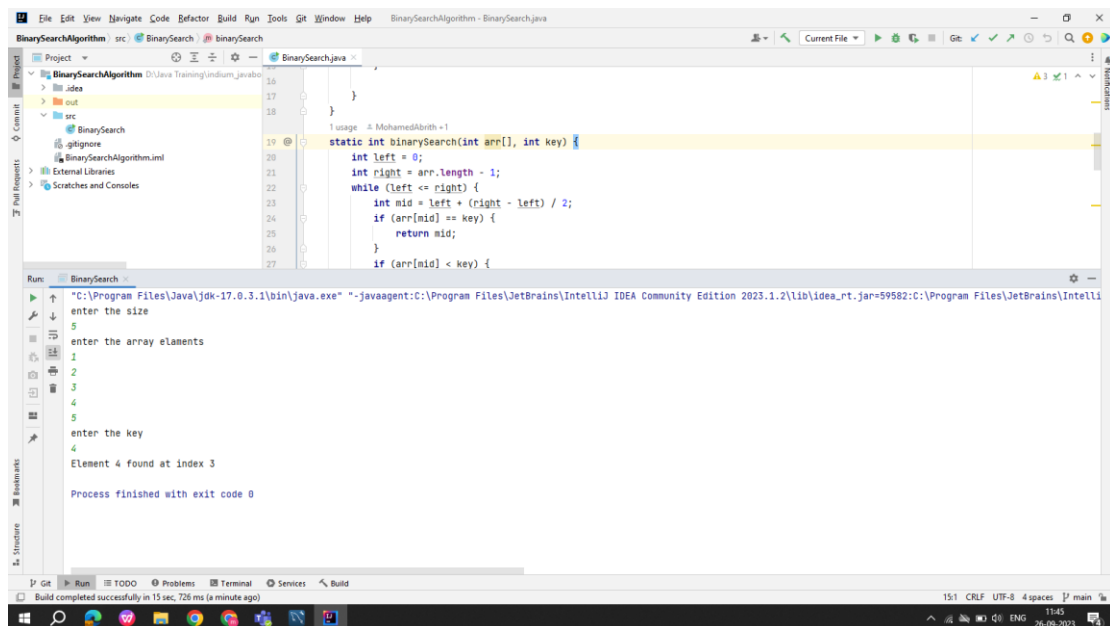
## CalculatorMethods



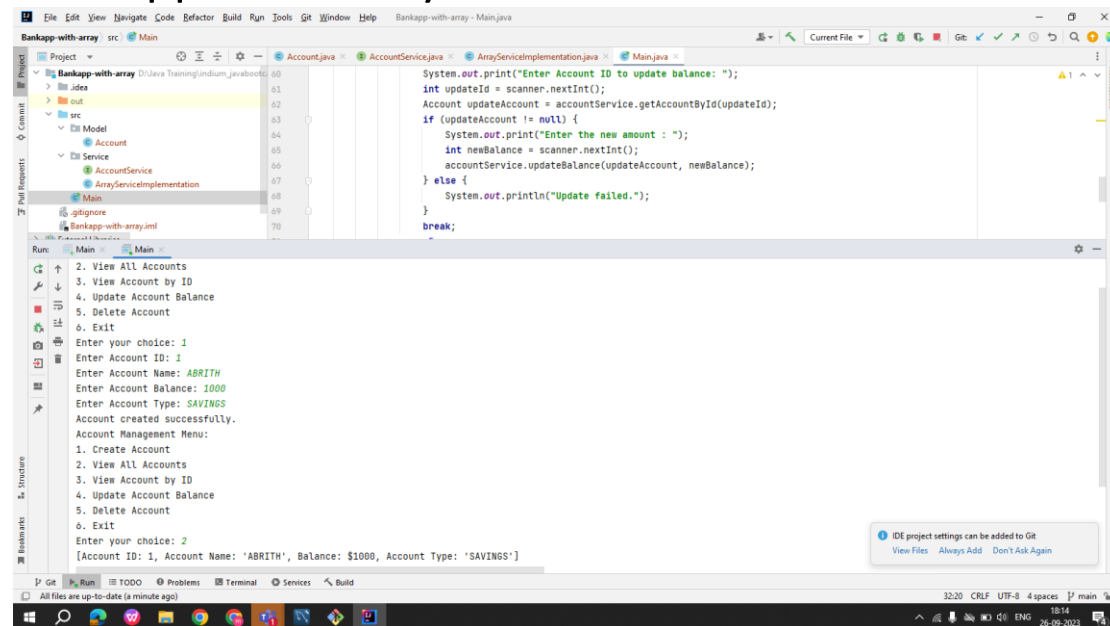
# Calculator



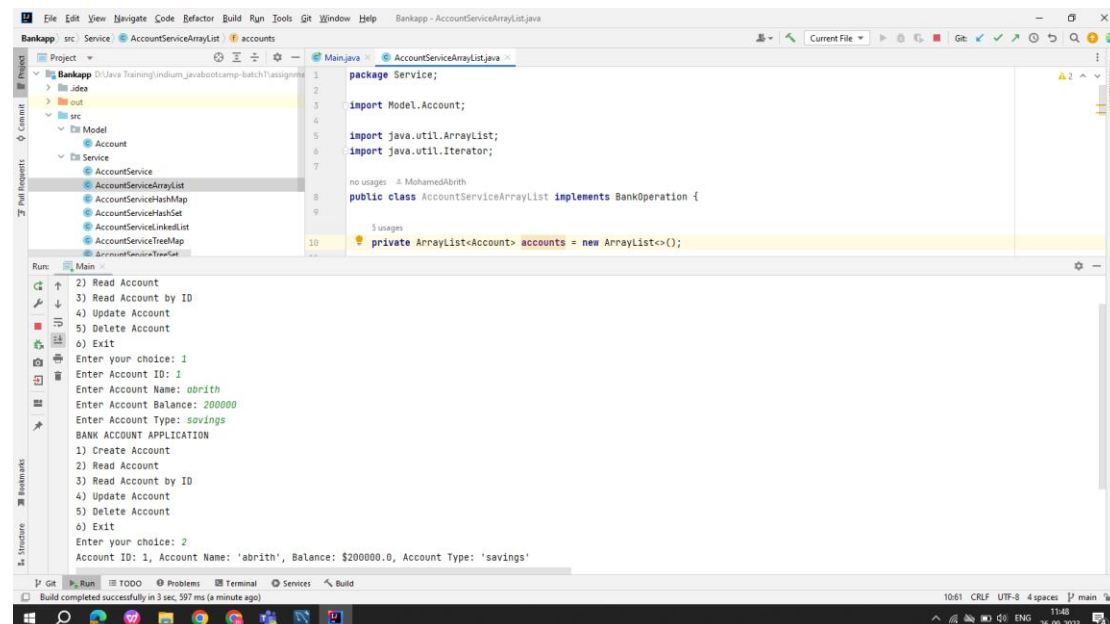
# BinarySearchAlgorithm



# Bankapp-with-array



# Bankapp-with-collection



# Bankapp-Import-Export

```
System.out.println("7. Import Data");
System.out.println("8. Export Data");
System.out.println("9. Exit");
System.out.print("Enter your choice: ");
choice = scanner.nextInt();

switch (choice) {
    case 1:
```

5. Delete Account  
6. Print Statistics  
7. Import Data  
8. Export Data  
9. Exit  
Enter your choice: 1  
Enter Account ID: 1  
Enter Account Name: abrith  
Enter Account Balance: 200000  
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: 1Account Name: abrithBalance: \$200000Account Type: savings

```
1. abrith,200000,savings
2.
```

7. Import Data  
8. Export Data  
9. Exit  
Enter your choice: 8  
Data exported 1  
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:

# Bankapp\_collection\_stats

The screenshot shows an IDE window for a project named "Bankapp\_collection\_stats". The project structure on the left includes a "src" folder with sub-packages "Model", "Service", and "Main". The "Main" package contains "BankOperation", "HashMapStats", and "Main" classes. The "Main.java" file is open, showing a menu-driven application. The console output shows the following sequence of events:

```
Enter Account Balance: 10000
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. Exit
Enter your choice: 0
Statistics:
No of accounts which have a balance more than 1 lac: 0
No of account by account type:
savings: 1
No of accounts by account type with sorting:
savings: 1
Avg balance by account type:
savings: 10000
Enter the partial name: abriith
Listing the account ids whose account name contains 'abriith':
```

The code in "Main.java" includes a switch statement for the menu options. The "case 4:" block is highlighted, showing the logic for updating an account balance. The code uses a "scanner" to read user input and a "HashMap" to store account data.

The screenshot shows an IDE with the following components:

- Menu Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Git, Window, Help.
- Project View (Left):** Shows the project structure for 'Bankapp-with-threads'. The 'src' folder contains 'Main.java', 'importStatistics', 'input', 'import.txt', 'output', and 'output.txt'.
- Main Window:** Displays the 'Main.java' file. The code defines a 'Runnable' interface and implements it in the 'Main' class. The 'main' method calls 'exportThread'.
- Run Console (Bottom):** Shows the output of the program. It includes a menu, user input, and thread execution details.

**Main.java Code:**

```

import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;

public class Main implements Runnable {
    private static ExecutorService executorService = Executors.newFixedThreadPool(10);

    @Override
    public void run() {
        // ... (code for menu and thread execution) ...
    }

    public void exportThread() {
        // ... (code for exporting thread) ...
    }
}

```

**Run Console Output:**

```

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: 7
Thread[pool-1-thread-1,main] is importing data
1.abrith,200000,deposit
Data Imported
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: 8
Thread[pool-1-thread-1,main] is exporting data
Data exported

```

The screenshot shows the IDE with the following components:

- Project Explorer:** Shows the project structure with folders for `src`, `input`, `output`, and `src`. The `src` folder contains `com.indium.bankapp`, which includes `Model`, `Account`, `Service`, `AccountService`, `AccountServiceHashMap`, `Main`, and `gIgnore`.
- Code Editor:** Displays the `AccountService.java` file. The `deleteAccount` method is implemented as follows:
 

```

@Override
public void deleteAccount(int id) { accountMap.remove(id); }
      
```
- Run Console:** Shows the program's output:
 

```

Enter Account Name: abrith
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: abrithBalance: $20000Account Type: savings
      
```

# Bankapp-with-JDBC

