

CURRENCY CONVERTOR AND MONEY

TRANSFERRING APPLICATION

Team details:

1. Name : KOPPARTHI TIRUMALA SRI PAVAN KUMAR LAXMI VINAY
(lead)
Reg no: 41110651
Branch: BE.CSE – C1
Phone : 9133997429
2. Name : KONA MAHESH KUMAR
Reg no: 41110644
Branch: BE.CSE – C1
Phone : 8074335186
3. Name : KANITHI JAYANTH BABU
Reg no: 41110570
Branch: BE.CSE – B5
Phone : 8897173918

Trainee : JEYANTH.V

GitHub link : [gitHub_Currencyconvertor](#)

DESCRIPTION

The application CURRENCY EXCHANGE is completely designed using core java concepts and it contains different methods for each operation , here we have created new method for each type of the conversion and also separate method for money transfer by collecting the details from the user.

In currency exchange methods first, we will get the input from user for which currency we should convert and alongside of the amount. Amount has a limit of 10000 and if it exceeds the value, it will display limit exceeded or if we enter the amount value other than numeric values it will display enter the correct value.

Currency exchange part consists of the input amount taken from the user and will display the current value of the exchangeable currency and initially it will calculate the amount that can be get by exchange and later next step it will calculate the commission amount and will display the exchangeable after subtracting the commission/tax amount from it.

Money transfer works by collecting the pre-requisites from the user like the name of the account holder, account number, bank name, branch name, IFSC code and the amount to be transferred.

By collecting these values and validating them in the system it will start the process of transferring the amount from exchange centre to persons bank account. At last of the process if the transfer is successful, it gives a message Monay transferred.

TOOLS USED

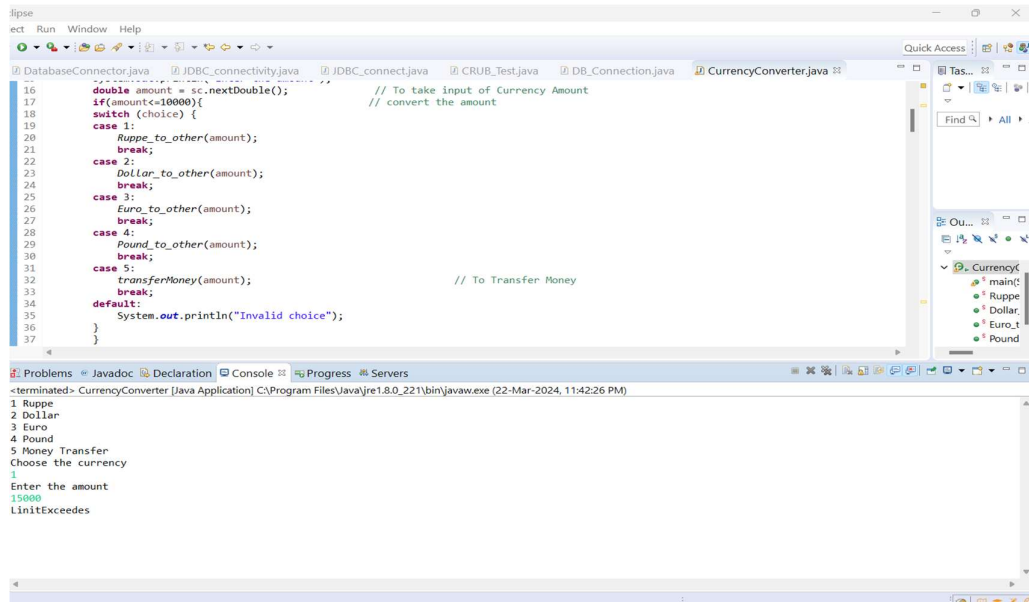
Language : Java

Compiler : Eclipse

OUTPUTS

When the input value exceeds the limit, it gives output as “LimitExceeds”

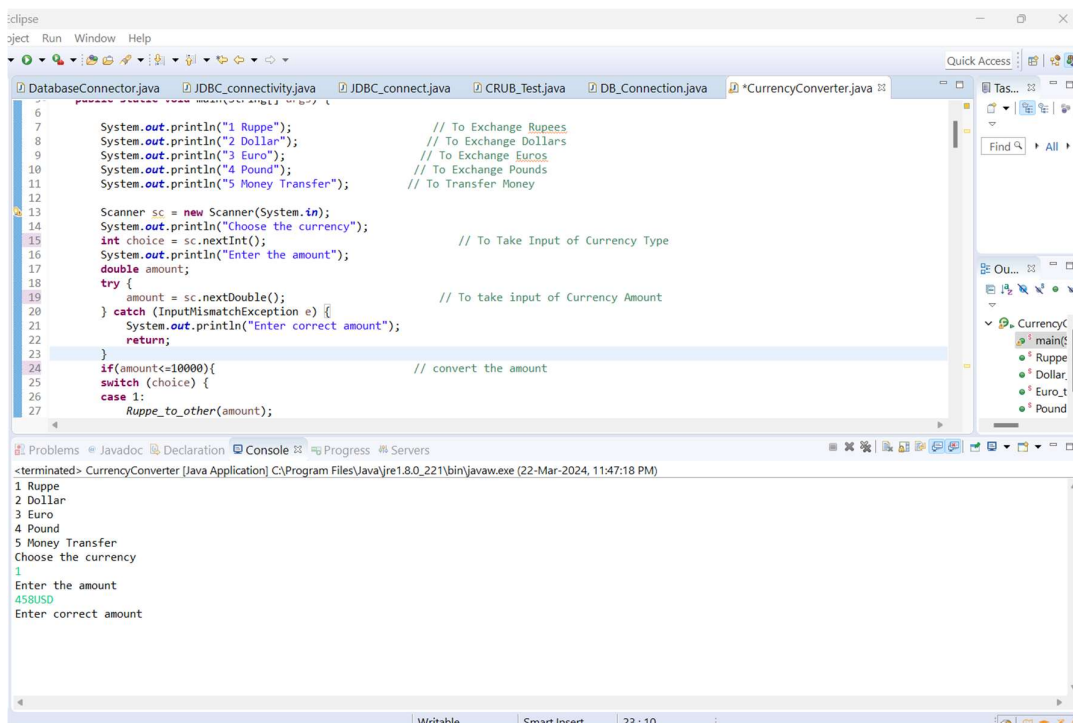
Here the limit is 15000 which is greater than declared limit.



```
DatabaseConnector.java  JDBC_connectivity.java  JDBC_connect.java  CRUB_Test.java  DB_Connection.java  CurrencyConverter.java
16  double amount = sc.nextDouble();          // To take input of Currency Amount
17  if(amount<=10000){                        // convert the amount
18  switch (choice) {
19  case 1:
20      Ruppe_to_other(amount);
21      break;
22  case 2:
23      Dollar_to_other(amount);
24      break;
25  case 3:
26      Euro_to_other(amount);
27      break;
28  case 4:
29      Pound_to_other(amount);
30      break;
31  case 5:
32      transferMoney(amount);                // To Transfer Money
33      break;
34  default:
35      System.out.println("Invalid choice");
36  }
37  }
```

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (22-Mar-2024, 11:42:26 PM)
1  Ruppe
2  Dollar
3  Euro
4  Pound
5  Money Transfer
Choose the currency
1
Enter the amount
15000
LimitExceeds
```

Here the value we entered is other than numeral so its giving enter the correct amount



```
DatabaseConnector.java  JDBC_connectivity.java  JDBC_connect.java  CRUB_Test.java  DB_Connection.java  *CurrencyConverter.java
6
7  System.out.println("1 Ruppe");              // To Exchange Rupees
8  System.out.println("2 Dollar");            // To Exchange Dollars
9  System.out.println("3 Euro");              // To Exchange Euros
10 System.out.println("4 Pound");             // To Exchange Pounds
11 System.out.println("5 Money Transfer");     // To Transfer Money
12
13 Scanner sc = new Scanner(System.in);
14 System.out.println("Choose the currency");
15 int choice = sc.nextInt();                  // To Take Input of Currency Type
16 System.out.println("Enter the amount");
17 double amount;
18 try {
19     amount = sc.nextDouble();                // To take input of Currency Amount
20 } catch (InputMismatchException e) {
21     System.out.println("Enter correct amount");
22     return;
23 }
24 if(amount<=10000){                          // convert the amount
25 switch (choice) {
26 case 1:
27     Ruppe_to_other(amount);
```

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (22-Mar-2024, 11:47:18 PM)
1  Ruppe
2  Dollar
3  Euro
4  Pound
5  Money Transfer
Choose the currency
1
Enter the amount
458USD
Enter correct amount
```

Selecting the valid testcases for conversion or transfer of currency

The screenshot shows an IDE with the `CurrencyConverter.java` file open. The code is a switch statement that handles different currency conversion and transfer cases. The console output shows the program running and prompting the user to choose a currency.

```
16 switch (choice) {
17     case 1:
18         Ruppee_to_other(amount);
19         break;
20     case 2:
21         Dollar_to_other(amount);
22         break;
23     case 3:
24         Euro_to_other(amount);
25         break;
26     case 4:
27         Pound_to_other(amount);
28         break;
29     case 5:
30         transferMoney(amount); // To Transfer Money
31         break;
32     default:
33         // Invalid choice
34 }
```

Console Output:

```
1 Ruppee
2 Dollar
3 Euro
4 Pound
5 Money Transfer
Choose the currency
```

If we enter other than above cases it will return Invalid case

The screenshot shows the same IDE with the `CurrencyConverter.java` file. The code is identical to the previous one, but the console output shows the program running and prompting the user to choose a currency. The user enters an invalid choice, and the program outputs "Invalid choice".

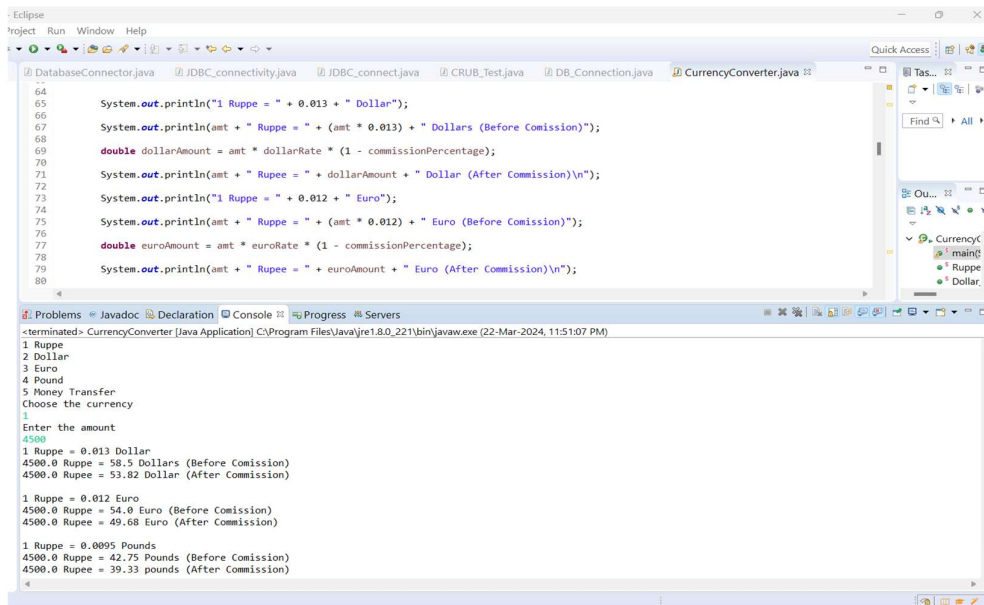
```
16 double amount = sc.nextDouble(); // To take input of Currency Amount
17 if (amount <= 10000) { // convert the amount
18     switch (choice) {
19         case 1:
20             Ruppee_to_other(amount);
21             break;
22         case 2:
23             Dollar_to_other(amount);
24             break;
25         case 3:
26             Euro_to_other(amount);
27             break;
28         case 4:
29             Pound_to_other(amount);
30             break;
31         case 5:
32             transferMoney(amount); // To Transfer Money
33             break;
34         default:
35             System.out.println("Invalid choice");
36     }
37 }
```

Console Output:

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (22-Mar-2024, 11:39:28 PM)
1 Ruppee
2 Dollar
3 Euro
4 Pound
5 Money Transfer
Choose the currency
12000
Enter the amount
1000
Invalid choice
```

Currency exchange outputs for all cases

Rupees to others



The screenshot shows the Eclipse IDE with the `CurrencyConverter.java` file open. The code defines exchange rates and commission percentages for Rupees to Dollars, Euros, and Pounds. The console output shows the results of running the application, where 4500 Rupees are converted to Dollars, Euros, and Pounds, showing both the amount before and after a 13% commission.

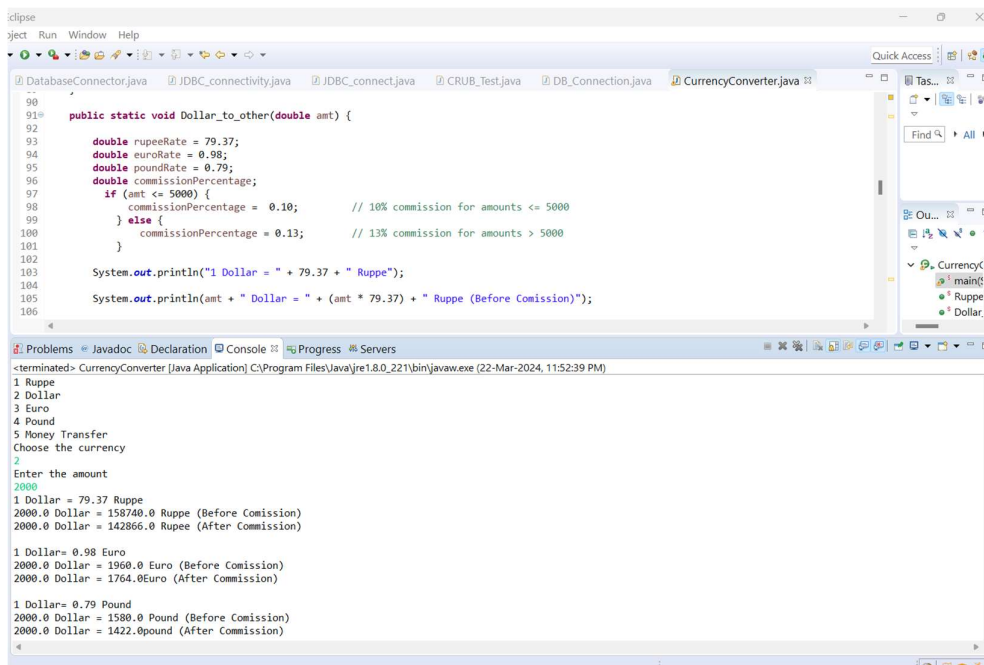
```
64      System.out.println("1 Rupee = " + 0.013 + " Dollar");
65
66      System.out.println(amt + " Rupee = " + (amt * 0.013) + " Dollars (Before Comission)");
67
68      double dollarAmount = amt * dollarRate * (1 - commissionPercentage);
69
70      System.out.println(amt + " Rupee = " + dollarAmount + " Dollar (After Comission)\n");
71
72      System.out.println("1 Rupee = " + 0.012 + " Euro");
73
74      System.out.println(amt + " Rupee = " + (amt * 0.012) + " Euro (Before Comission)");
75
76      double euroAmount = amt * euroRate * (1 - commissionPercentage);
77
78      System.out.println(amt + " Rupee = " + euroAmount + " Euro (After Comission)\n");
79
80
```

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\java.exe (22-Mar-2024, 11:51:07 PM)
1 Rupee
2 Dollar
3 Euro
4 Pound
5 Money Transfer
Choose the currency
1
Enter the amount
4500
1 Rupee = 0.013 Dollar
4500.0 Rupee = 58.5 Dollars (Before Comission)
4500.0 Rupee = 53.82 Dollar (After Comission)

1 Rupee = 0.012 Euro
4500.0 Rupee = 54.0 Euro (Before Comission)
4500.0 Rupee = 49.68 Euro (After Comission)

1 Rupee = 0.0095 Pounds
4500.0 Rupee = 42.75 Pounds (Before Comission)
4500.0 Rupee = 39.33 pounds (After Comission)
```

Dollars to others



The screenshot shows the Eclipse IDE with the `CurrencyConverter.java` file open. The `Dollar_to_other` method is highlighted, which uses conditional logic to apply different commission percentages based on the amount. The console output shows the results of running the application, where 2000 Dollars are converted to Rupees, Euros, and Pounds, showing both the amount before and after a commission (10% for amounts ≤ 5000, 13% for amounts > 5000).

```
90
91 public static void Dollar_to_other(double amt) {
92
93     double rupeeRate = 79.37;
94     double euroRate = 0.98;
95     double poundRate = 0.79;
96     double commissionPercentage;
97     if (amt <= 5000) {
98         commissionPercentage = 0.10; // 10% commission for amounts <= 5000
99     } else {
100         commissionPercentage = 0.13; // 13% commission for amounts > 5000
101     }
102
103     System.out.println("1 Dollar = " + 79.37 + " Rupee");
104
105     System.out.println(amt + " Dollar = " + (amt * 79.37) + " Rupee (Before Comission)");
106
```

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\java.exe (22-Mar-2024, 11:52:39 PM)
1 Rupee
2 Dollar
3 Euro
4 Pound
5 Money Transfer
Choose the currency
2
Enter the amount
2000
1 Dollar = 79.37 Rupee
2000.0 Dollar = 158740.0 Rupee (Before Comission)
2000.0 Dollar = 142866.0 Rupee (After Comission)

1 Dollar= 0.98 Euro
2000.0 Dollar = 1960.0 Euro (Before Comission)
2000.0 Dollar = 1764.0Euro (After Comission)

1 Dollar= 0.79 Pound
2000.0 Dollar = 1580.0 Pound (Before Comission)
2000.0 Dollar = 1422.0pound (After Comission)
```

Euros to others

The screenshot shows the Eclipse IDE with the `CurrencyConverter.java` file open. The `Euro_to_other` method is visible, which calculates conversions for Rupee, Dollar, and Pound based on fixed rates and a commission percentage. The console output shows the program's execution flow, including menu navigation and conversion results for 9000 Euros.

```
public static void Euro_to_other(double amt) {
    double rupeeRate = 80.85;
    double dollarRate = 1.02;
    double poundRate = 0.86;
    double commissionPercentage;
    if (amt <= 5000) {
        commissionPercentage = 0.12; // 12% commission for amounts <= 5000
    } else {
        commissionPercentage = 0.15; // 15% commission for amounts > 5000
    }

    System.out.println("1 Euro = " + 80.85 + " Rupee");
    System.out.println(amt + " Euro = " + (amt * 80.85) + " Rupee (Before Comission)");
    double rupeeAmount = amt * rupeeRate * (1 - commissionPercentage);
    System.out.println(amt + " Euro = " + rupeeAmount + " Rupee (After Comission)");

    System.out.println("1 Euro = " + 1.02 + " Dollar");
    System.out.println(amt + " Euro = " + (amt * 1.02) + " Dollar (Before Comission)");
    double dollarAmount = amt * dollarRate * (1 - commissionPercentage);
    System.out.println(amt + " Euro = " + dollarAmount + " Dollar (After Comission)");

    System.out.println("1 Euro = " + 0.86 + " Pound");
    System.out.println(amt + " Euro = " + (amt * 0.86) + " Pound (Before Comission)");
    double poundAmount = amt * poundRate * (1 - commissionPercentage);
    System.out.println(amt + " Euro = " + poundAmount + " Pounds (After Comission)");
}
```

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (22-Mar-2024, 11:53:40 PM)
1 Rupee
2 Dollar
3 Euro
4 Pound
5 Money Transfer
Choose the currency
3
Enter the amount
9000
1 Euro = 80.85 Rupee
9000.0 Euro = 727650.0 Rupee (Before Comission)
9000.0 Euro = 618502.5 Rupee (After Comission)

1 Euro = 1.02 Dollar
9000.0 Euro = 9180.0 Dollar (Before Comission)
9000.0Euro = 7803.0 Dollar (After Comission)

1 Euro = 0.86 Pound
9000.0 Euro = 7740.0 Pound (Before Comission)
9000.0Euro = 6579.0 pounds (After Comission)
```

Pounds to others

The screenshot shows the Eclipse IDE with the `CurrencyConverter.java` file open. The `Pound_to_other` method is visible, which calculates conversions for Rupees, Euro, and Dollars based on fixed rates and a commission percentage. The console output shows the program's execution flow, including menu navigation and conversion results for 1500 Pounds.

```
public static void Pound_to_other(double amt) {
    double rupeeRate = 105.24;
    double euroRate = 1.17;
    double dollarRate = 1.26;
    double commissionPercentage;
    if (amt <= 5000) {
        commissionPercentage = 0.05; // 5% commission for amounts <= 5000
    } else {
        commissionPercentage = 0.06; // 6% commission for amounts > 5000
    }

    System.out.println("1 Pound = " + 105.24 + " Rupees");
    System.out.println(amt + " Pound = " + (amt * 105.24) + " Rupees (Before Comission)");
    double rupeeAmount = amt * rupeeRate * (1 - commissionPercentage);
    System.out.println(amt + " Pounds = " + rupeeAmount + " Rupees (After Comission)");

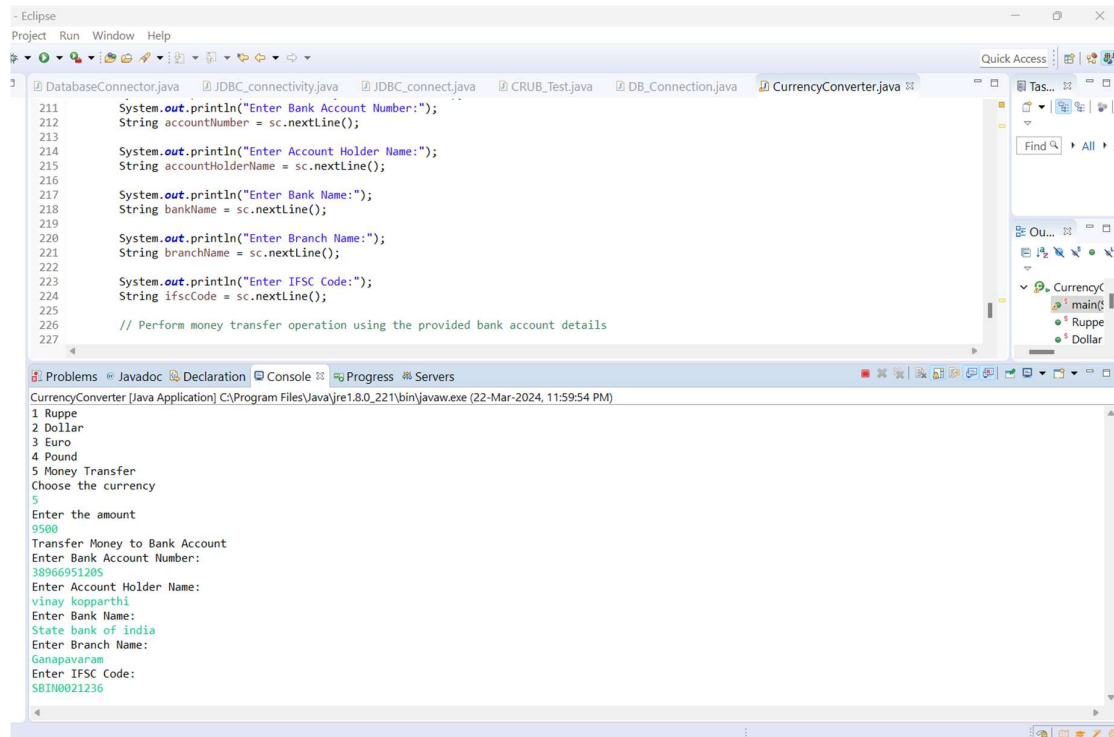
    System.out.println("1 Pound = " + 1.17 + " Euro");
    System.out.println(amt + " Pound = " + (amt * 1.17) + " Euro (Before Comission)");
    double euroAmount = amt * euroRate * (1 - commissionPercentage);
    System.out.println(amt + " Pounds = " + euroAmount + " Euros (After Comission)");

    System.out.println("1 Pound = " + 1.26 + " Dollars");
    System.out.println(amt + " Pound = " + (amt * 1.26) + " Dollars (Before Comission)");
    double dollarAmount = amt * dollarRate * (1 - commissionPercentage);
    System.out.println(amt + " Pounds = " + dollarAmount + " Euros (After Comission)");
}
```

```
<terminated> CurrencyConverter [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (22-Mar-2024, 11:54:41 PM)
1 Rupee
2 Dollar
3 Euro
4 Pound
5 Money Transfer
Choose the currency
4
Enter the amount
1500
1 Pound = 105.24 Rupees
1500.0 Pound = 157860.0 Rupees (Before Comission)
1500.0Pounds = 149967.0 Rupees (After Comission)
1 Pound = 1.17 Euro
1500.0 Pound = 1755.0 Euro (Before Comission)
1500.0Pounds = 1667.25 Euros (After Comission)

1 Pound = 1.26 Dollars
1500.0 Pound = 1890.0 Dollars (Before Comission)
1500.0Pounds = 1795.5 Euros (After Comission)
```

Collecting details for money transfer



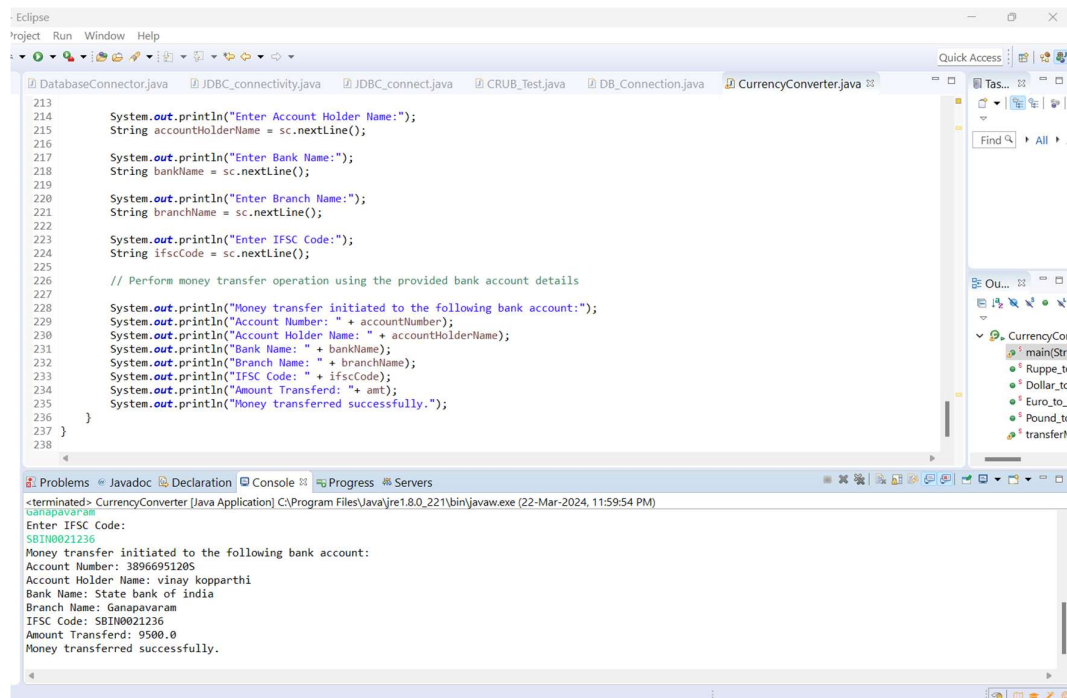
The screenshot shows the Eclipse IDE with the `CurrencyConverter.java` file open. The code prompts the user to enter bank account details. The console output shows the user's input for each prompt.

```
211 System.out.println("Enter Bank Account Number:");
212 String accountNumber = sc.nextLine();
213
214 System.out.println("Enter Account Holder Name:");
215 String accountHolderName = sc.nextLine();
216
217 System.out.println("Enter Bank Name:");
218 String bankName = sc.nextLine();
219
220 System.out.println("Enter Branch Name:");
221 String branchName = sc.nextLine();
222
223 System.out.println("Enter IFSC Code:");
224 String ifscCode = sc.nextLine();
225
226 // Perform money transfer operation using the provided bank account details
227
```

Console Output:

```
1 Ruppee
2 Dollar
3 Euro
4 Pound
5 Money Transfer
Choose the currency
5
Enter the amount
9500
Transfer Money to Bank Account
Enter Bank Account Number:
38966951205
Enter Account Holder Name:
vinay kopparthi
Enter Bank Name:
State bank of india
Enter Branch Name:
Ganapavaram
Enter IFSC Code:
SBIIN0021236
```

Transaction successful message



The screenshot shows the Eclipse IDE with the `CurrencyConverter.java` file open. The code now includes a final message indicating the successful completion of the money transfer. The console output shows the user's input and the final success message.

```
213
214 System.out.println("Enter Account Holder Name:");
215 String accountHolderName = sc.nextLine();
216
217 System.out.println("Enter Bank Name:");
218 String bankName = sc.nextLine();
219
220 System.out.println("Enter Branch Name:");
221 String branchName = sc.nextLine();
222
223 System.out.println("Enter IFSC Code:");
224 String ifscCode = sc.nextLine();
225
226 // Perform money transfer operation using the provided bank account details
227
228 System.out.println("Money transfer initiated to the following bank account:");
229 System.out.println("Account Number: " + accountNumber);
230 System.out.println("Account Holder Name: " + accountHolderName);
231 System.out.println("Bank Name: " + bankName);
232 System.out.println("Branch Name: " + branchName);
233 System.out.println("IFSC Code: " + ifscCode);
234 System.out.println("Amount Transferred: " + amt);
235 System.out.println("Money transferred successfully.");
236 }
237 }
238
```

Console Output:

```
<terminated> CurrencyConverter (Java Application) C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (22-Mar-2024, 11:59:54 PM)
ganapavaram
Enter IFSC Code:
SBIIN0021236
Money transfer initiated to the following bank account:
Account Number: 38966951205
Account Holder Name: vinay kopparthi
Bank Name: State bank of india
Branch Name: Ganapavaram
IFSC Code: SBIIN0021236
Amount Transferred: 9500.0
Money transferred successfully.
```

RESULT & CONCLUSION

Currency exchange and money transferring application has successfully executed using core java concepts .

Thus the application accurately exchanges the currency into one another and will transfer the currency from exchange centre to personal bank account.