```java
package emiCal;

import java.time.LocalDate;

public class AccountDetails {
    private long accountNo;
    private String accountHolderName;
    private double accountBalance;
    private String address;
    private LocalDate dateOfBirth;
    public long getAccountNo() {
        return accountNo;
    }
    public String getAccountHolderName() {
        return accountHolderName;
    }
    public double getAccountBalance() {
        return accountBalance;
    }
    public String getAddress() {
        return address;
    }
    public LocalDate getDateOfBirth() {
        return dateOfBirth;
    }
    public void setAccountNo(long accountNo) {
        this.accountNo = accountNo;
    }
    public void setAccountHolderName(String accountHolderName) {
        this.accountHolderName = accountHolderName;
    }
    public void setAccountBalance(double accountBalance) {
        this.accountBalance = accountBalance;
    }
    public void setAddress(String address) {
        this.address = address;
```

```
35    public void setAddress(String address) {
36        this.address = address;
37    }
38    public void setDateOfBirth(LocalDate dateOfBirth) {
39        this.dateOfBirth = dateOfBirth;
40    }
41
42
43    public AccountDetails() {
44        super();
45    }
46    public AccountDetails(long accountNo, String accountHolderName, double accountBalance, String address) {
47        super();
48        this.accountNo = accountNo;
49        this.accountHolderName = accountHolderName;
50        this.accountBalance = accountBalance;
51        this.address = address;
52    }
53    @Override
54    public String toString() {
55        return "AccountDetails [accountNo=" + accountNo + ", accountHolderName=" + accountHolderName
56                + ", accountBalance=" + accountBalance + ", address=" + address + "]";
57    }
58
59
60
61
62
63 }
```

Writable | Smart Insert | 30 : 52 : 740

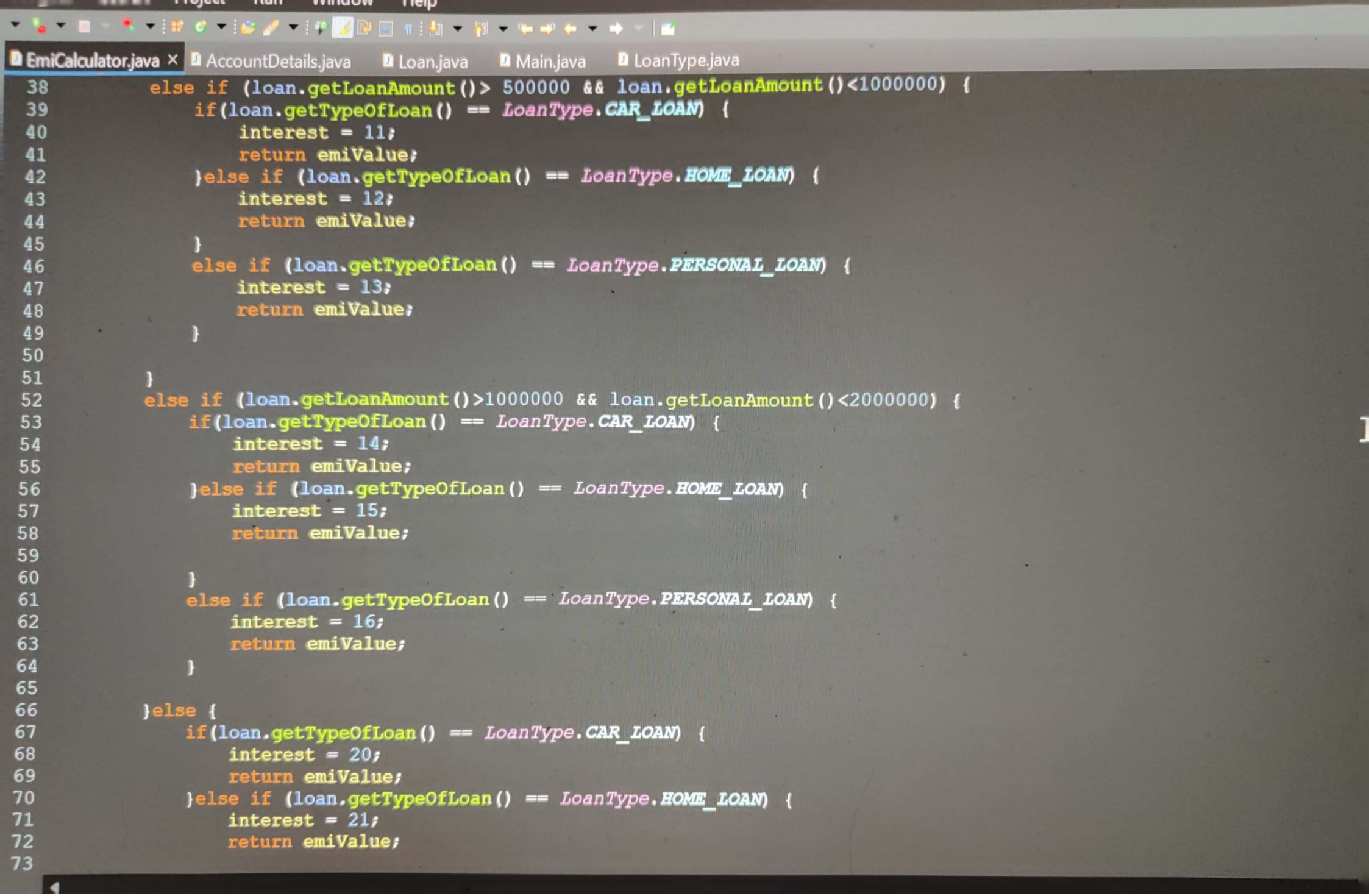
Search

```java
public class Loan {
private LoanType typeOfLoan;
private double loanAmount;
private int noOfMonths;
public LoanType getTypeOfLoan() {
    return typeOfLoan;
}
public double getLoanAmount() {
    return loanAmount;
}
public int getNoOfMonths() {
    return noOfMonths;
}
public void setTypeOfLoan(LoanType typeOfLoan) {
    this.typeOfLoan = typeOfLoan;
}
public void setLoanAmount(double loanAmount) {
    this.loanAmount = loanAmount;
}
public void setNoOfMonths(int noOfMonths) {
    this.noOfMonths = noOfMonths;
}
public Loan(LoanType typeOfLoan, double loanAmount, int noOfMonths) {

    this.typeOfLoan = typeOfLoan;
    this.loanAmount = loanAmount;
    this.noOfMonths = noOfMonths;
}
public Loan() {

}
@Override
public String toString() {
    return "Loan [typeOfLoan=" + typeOfLoan + ", loanAmount=" + loanAmount + ", noOfMonths=" + noOfMonths + "]";
}

```

```
2
3 public enum LoanType {
4 CAR_LOAN, HOME_LOAN, PERSONAL_LOAN;
5 }
6
```

EmiCalculator.java | AccountDetails.java | Loan.java | Main.java | LoanType.java

```java
package emiCal;

import java.time.LocalDate;

public class EmiCalculator {
    public static Boolean isEligibleForLoan(Loan loan,AccountDetails acc) {

     if(acc.getAccountBalance() < (25/100)*loan.getLoanAmount()) {
         return false;

     }else {
        return true;
    }

    }

    public static double calculateEmi(Loan loan) {

        Integer interest = 0;
        double emiValue = (loan.getLoanAmount()/loan.getNoOfMonths())+(interest*100)/loan.getLoanAmount();
        if(loan.getLoanAmount()< 500000) {
            if(loan.getTypeOfLoan() == LoanType.CAR_LOAN) {
                interest = 8;
                return emiValue;
            }else if (loan.getTypeOfLoan() == LoanType.HOME_LOAN) {
                interest = 9;
                return emiValue;
            }
            else if (loan.getTypeOfLoan() == LoanType.PERSONAL_LOAN) {
                interest = 10;
                return emiValue;
            }

        }
        else if (loan.getLoanAmount()> 500000 && loan.getLoanAmount()<1000000) {
            if(loan.getTypeOfLoan() == LoanType.CAR_LOAN) {
```


EmiCalculator.java
AccountDetails.java
Loan.java
Main.java
LoanType.java
38 else if (loan.getLoanAmount()> 500000 && loan.getLoanAmount()<1000000) {
39 if(loan.getTypeOfLoan() == LoanType.CAR_LOAN) {
40 interest = 11;
41 return emiValue;
42 }else if (loan.getTypeOfLoan() == LoanType.HOME_LOAN) {
43 interest = 12;
44 return emiValue;
45 }
46 else if (loan.getTypeOfLoan() == LoanType.PERSONAL_LOAN) {
47 interest = 13;
48 return emiValue;
49 }
50
51 }
52 else if (loan.getLoanAmount()>1000000 && loan.getLoanAmount()<2000000) {
53 if(loan.getTypeOfLoan() == LoanType.CAR_LOAN) {
54 interest = 14;
55 return emiValue;
56 }else if (loan.getTypeOfLoan() == LoanType.HOME_LOAN) {
57 interest = 15;
58 return emiValue;
59
60 }
61 else if (loan.getTypeOfLoan() == LoanType.PERSONAL_LOAN) {
62 interest = 16;
63 return emiValue;
64 }
65
66 }else {
67 if(loan.getTypeOfLoan() == LoanType.CAR_LOAN) {
68 interest = 20;
69 return emiValue;
70 }else if (loan.getTypeOfLoan() == LoanType.HOME_LOAN) {
71 interest = 21;
72 return emiValue;
73

Calculator.java | AccountDetails.java | Loan.java | Main.java | LoanType.java

```
            return emiValue;

        }
        else if (loan.getTypeOfLoan() == LoanType.PERSONAL_LOAN) {
            interest = 22;
            return emiValue;
        }
    }
    return emiValue;


}
public static Boolean isValidAccountHolderDOB(LocalDate dateOfBirth) {
    LocalDate date = LocalDate.of(2022, 12, 20);
    int age = Period.between( dateOfBirth,date).getYears();
    if(age >21 && age <41) {
        return true;
    }else {
        return false;
    }

}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    AccountDetails acc1= new AccountDetails(sc.nextLong(), sc.next(), sc.nextDouble(), sc.next());
    String dateOfBirthInString=sc.next();
    DateTimeFormatter formattermatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
    LocalDate dateOfBirth= LocalDate.parse(dateOfBirthInString,formattermatter);
    Loan loan = new Loan(LoanType.valueOf(sc.next().toUpperCase()), sc.nextDouble(), sc.nextInt());
    if(isValidAccountHolderDOB(dateOfBirth) && isEligibleForLoan(loan, acc1)) {
        System.out.printf("EMI is: %.2f%n" ,calculateEmi(loan));
    }else {
        System.out.println("Sorry!! you are not eligible to get a loan");
    }
    sc.close();
```

EmiCalculator.java | AccountDetails.java | Loan.java | Main.java | LoanType.java

```
86          int age = Period.between( dateOfBirth,date).getYears();
87          if(age >21 && age <41) {
88              return true;
89          }else {
90              return false;
91          }
92
93      }
94      public static void main(String[] args) {
95          Scanner sc = new Scanner(System.in);
96          AccountDetails accl= new AccountDetails(sc.nextLong(), sc.next(), sc.nextDouble(), sc.next());
97          String dateOfBirthInString=sc.next();
98          DateTimeFormatter formattermatter = DateTimeFormatter.ofPattern("d/MM/yyyy");
99          LocalDate dateOfBirth= LocalDate.parse(dateOfBirthInString,formattermatter);
100         Loan loan = new Loan(LoanType.valueOf(sc.next().toUpperCase()), sc.nextDouble(), sc.nextInt());
101         if(isValidAccountHolderDOB(dateOfBirth) && isEligibleForLoan(loan, acc1)) {
102             System.out.printf("EMI is: %.2f%n" ,calculateEmi(loan));
103         }else {
104             System.out.println("Sorry!! you are not eligible to get a loan");
105         }
106         sc.close();
107
108     }
109
110
111 }
112
113
114 //10001001
115 //john
116 //200000
117 //Boston
118 //12/01/2005
119 //Car_Loan
120 //15000
121 //72
```