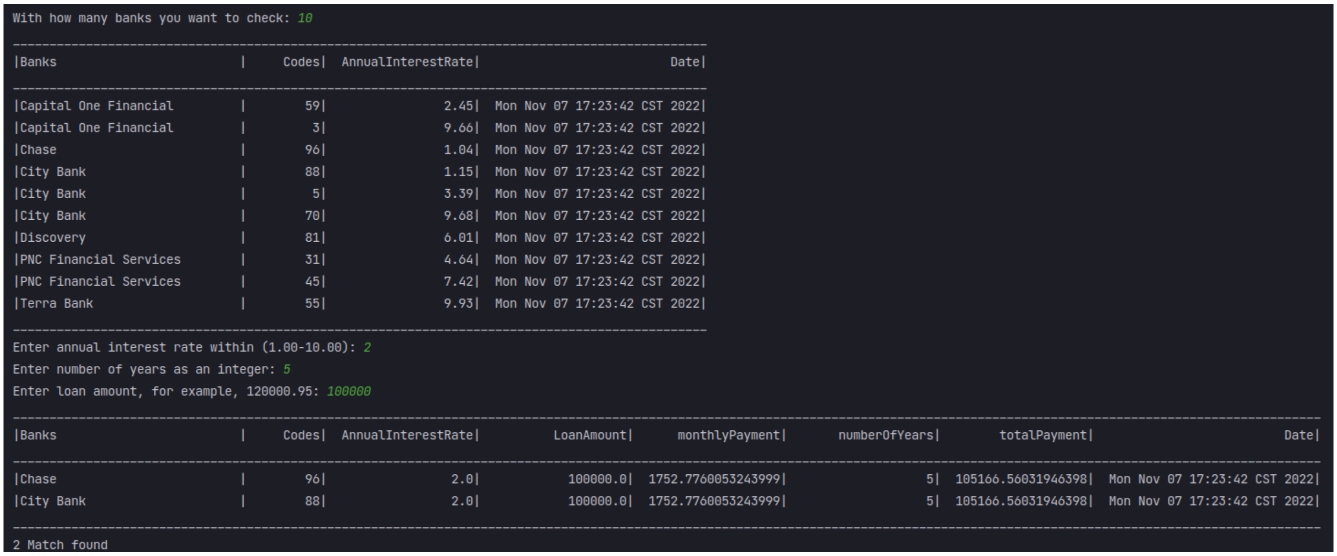
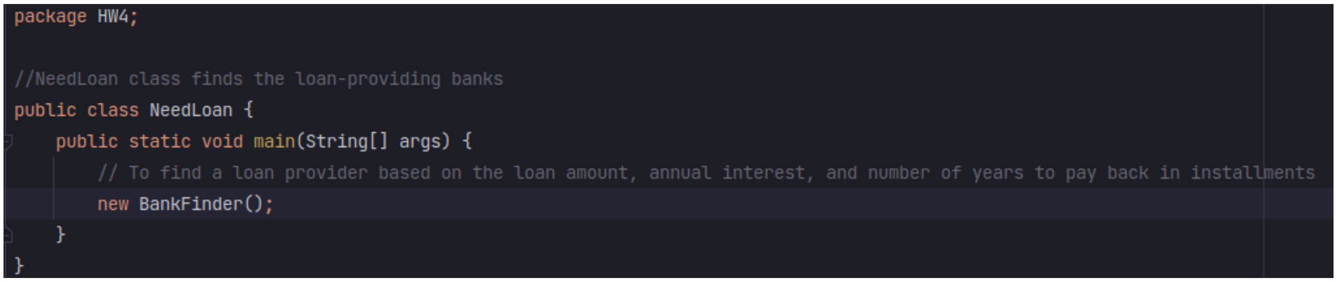
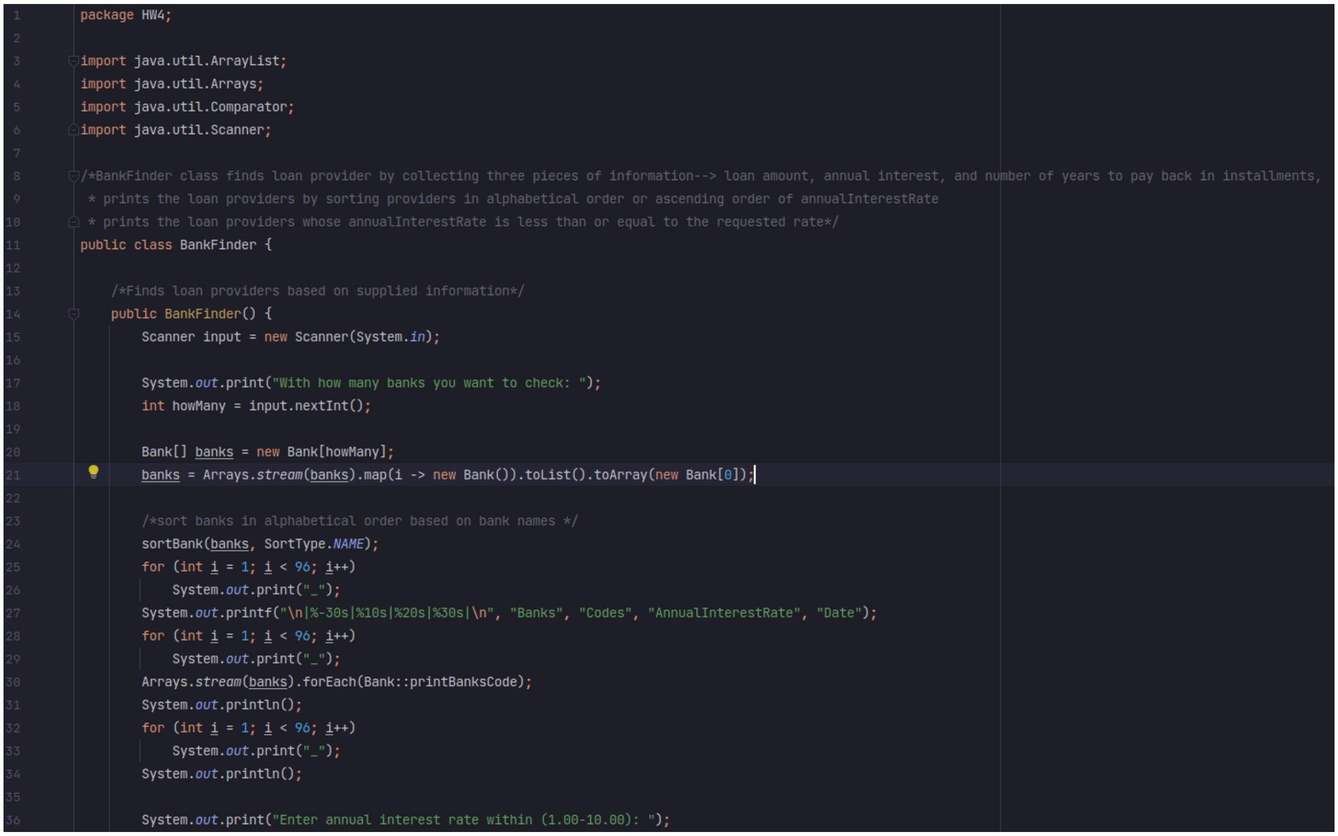


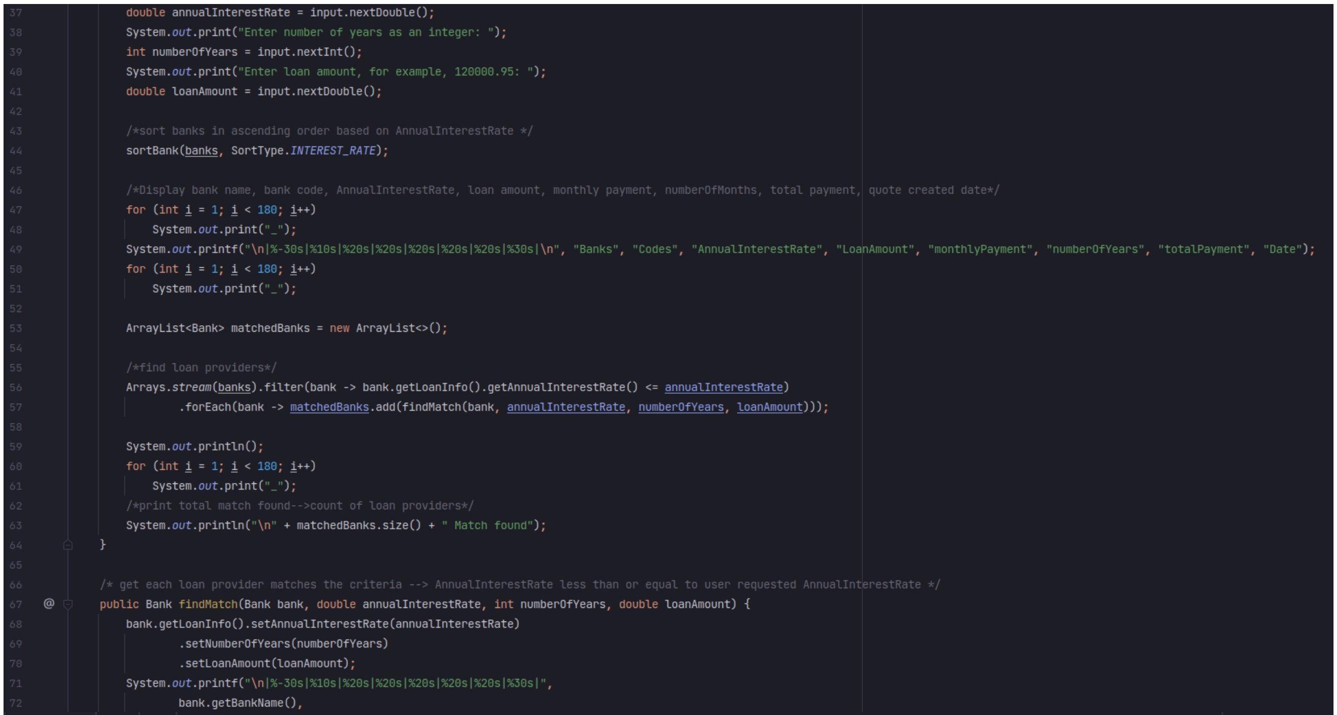
```
With how many banks you want to check: 10

-----------------------------------------------------------------------------------------------
|Banks                         |     Codes|  AnnualInterestRate|                          Date|
-----------------------------------------------------------------------------------------------
|Capital One Financial         |        59|                2.45|  Mon Nov 07 17:23:42 CST 2022|
|Capital One Financial         |         3|                9.66|  Mon Nov 07 17:23:42 CST 2022|
|Chase                         |        96|                1.04|  Mon Nov 07 17:23:42 CST 2022|
|City Bank                     |        88|                1.15|  Mon Nov 07 17:23:42 CST 2022|
|City Bank                     |         5|                3.39|  Mon Nov 07 17:23:42 CST 2022|
|City Bank                     |        70|                9.68|  Mon Nov 07 17:23:42 CST 2022|
|Discovery                     |        81|                6.01|  Mon Nov 07 17:23:42 CST 2022|
|PNC Financial Services        |        31|                4.64|  Mon Nov 07 17:23:42 CST 2022|
|PNC Financial Services        |        45|                7.42|  Mon Nov 07 17:23:42 CST 2022|
|Terra Bank                    |        55|                9.93|  Mon Nov 07 17:23:42 CST 2022|
-----------------------------------------------------------------------------------------------
Enter annual interest rate within (1.00-10.00): 2
Enter number of years as an integer: 5
Enter loan amount, for example, 120000.95: 100000

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|Banks                         |     Codes|  AnnualInterestRate|          LoanAmount|      monthlyPayment|       numberOfYears|        totalPayment|                          Date|
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|Chase                         |        96|                 2.0|            100000.0|  1752.7760053243999|                   5|  105166.56031946398|  Mon Nov 07 17:23:42 CST 2022|
|City Bank                     |        88|                 2.0|            100000.0|  1752.7760053243999|                   5|  105166.56031946398|  Mon Nov 07 17:23:42 CST 2022|
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
2 Match found
```



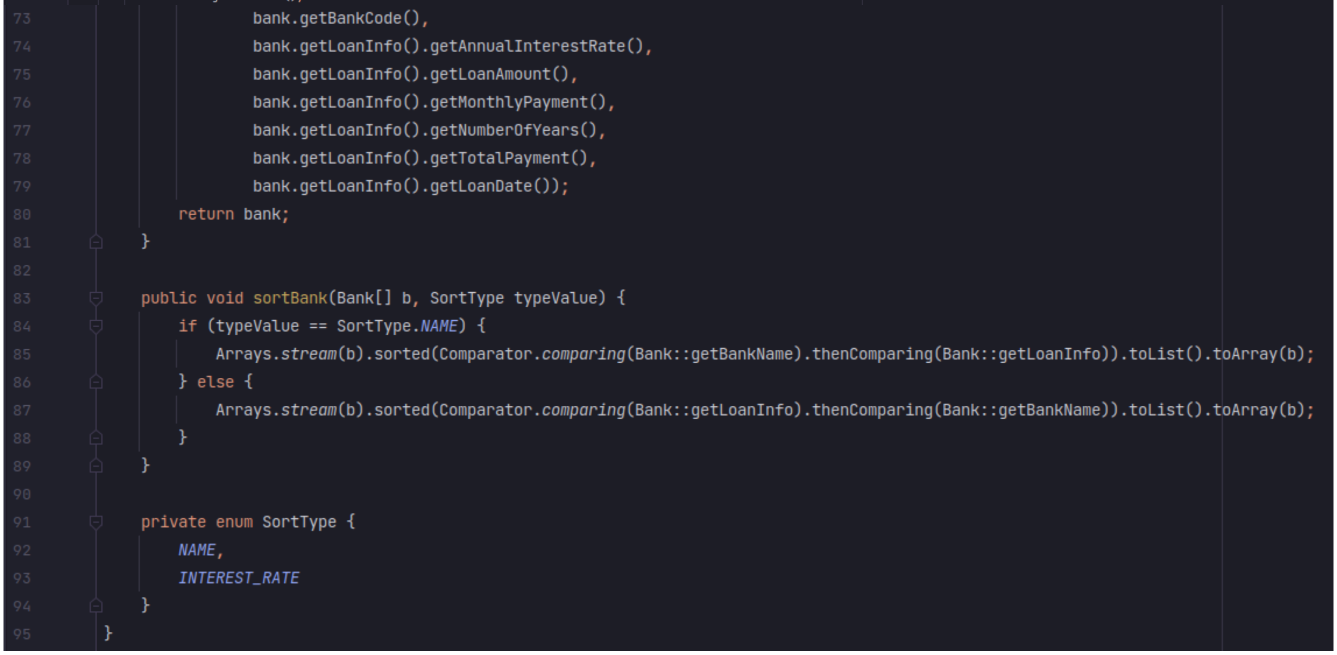
```java
package HW4;

//NeedLoan class finds the loan-providing banks
public class NeedLoan {
    public static void main(String[] args) {
        // To find a loan provider based on the loan amount, annual interest, and number of years to pay back in installments
        new BankFinder();
    }
}
```



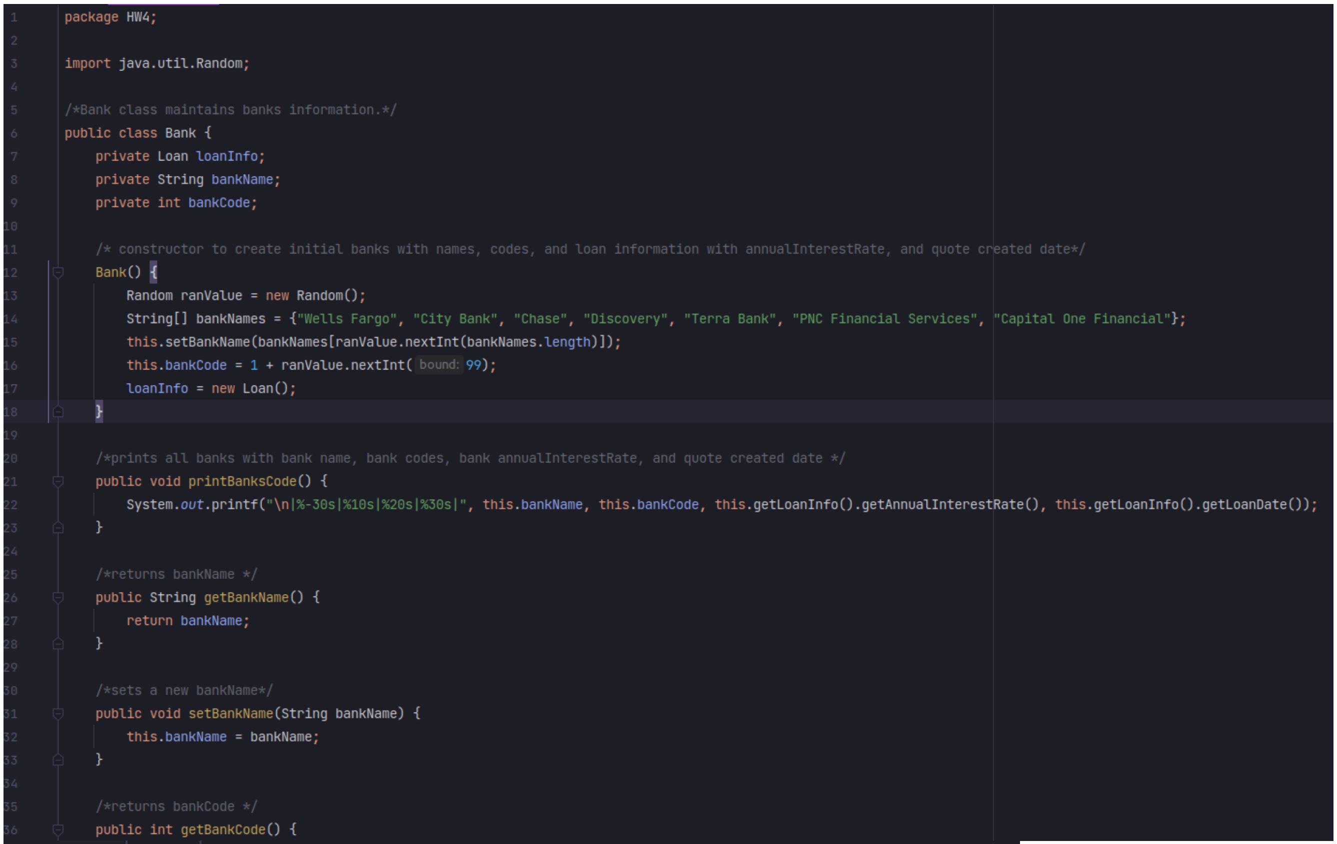
```java
package HW4;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Comparator;
import java.util.Scanner;

/*BankFinder class finds loan provider by collecting three pieces of information--> loan amount, annual interest, and number of years to pay back in installments,
 * prints the loan providers by sorting providers in alphabetical order or ascending order of annualInterestRate
 * prints the loan providers whose annualInterestRate is less than or equal to the requested rate*/
public class BankFinder {

    /*Finds loan providers based on supplied information*/
    public BankFinder() {
        Scanner input = new Scanner(System.in);

        System.out.print("With how many banks you want to check: ");
        int howMany = input.nextInt();

        Bank[] banks = new Bank[howMany];
        banks = Arrays.stream(banks).map(i -> new Bank()).toList().toArray(new Bank[0]);

        /*sort banks in alphabetical order based on bank names */
        sortBank(banks, SortType.NAME);
        for (int i = 1; i < 96; i++)
            System.out.print("_");
        System.out.printf("\n|%-30s|%10s|%20s|%30s|\n", "Banks", "Codes", "AnnualInterestRate", "Date");
        for (int i = 1; i < 96; i++)
            System.out.print("_");
        Arrays.stream(banks).forEach(Bank::printBanksCode);
        System.out.println();
        for (int i = 1; i < 96; i++)
            System.out.print("_");
        System.out.println();

        System.out.print("Enter annual interest rate within (1.00-10.00): ");
```



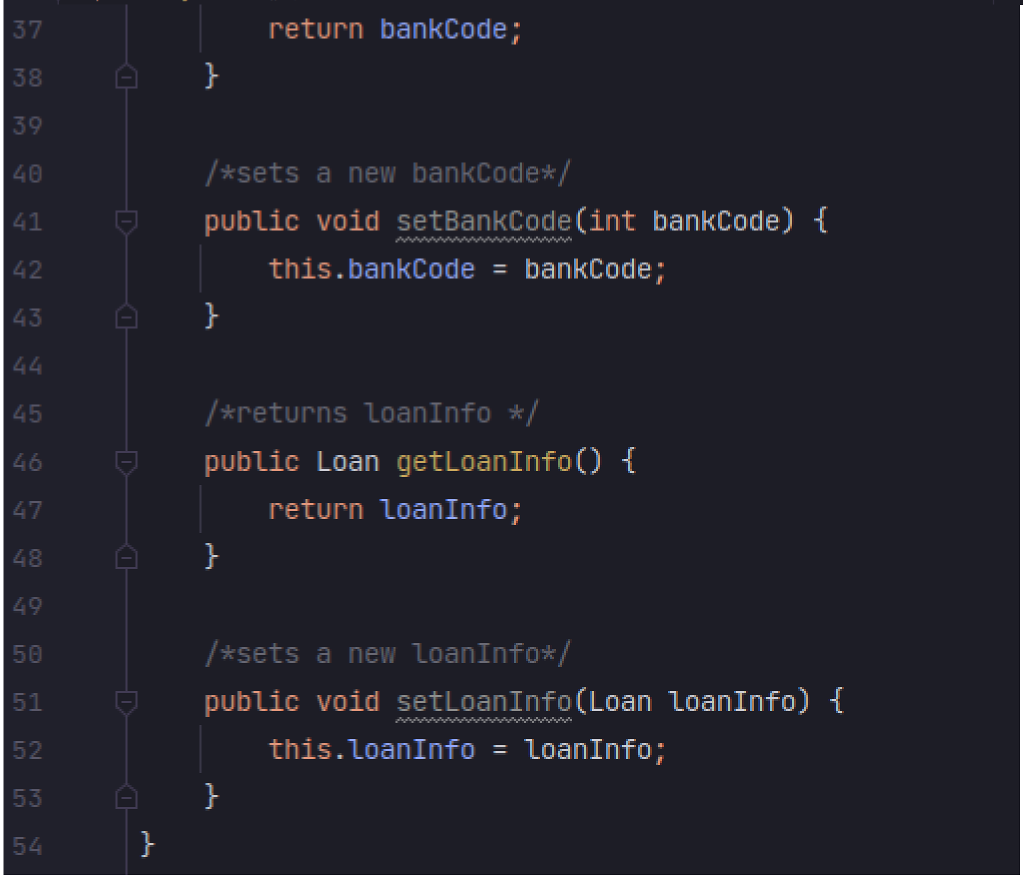
```java
        double annualInterestRate = input.nextDouble();
        System.out.print("Enter number of years as an integer: ");
        int numberOfYears = input.nextInt();
        System.out.print("Enter loan amount, for example, 120000.95: ");
        double loanAmount = input.nextDouble();

        /*sort banks in ascending order based on AnnualInterestRate */
        sortBank(banks, SortType.INTEREST_RATE);

        /*Display bank name, bank code, AnnualInterestRate, loan amount, monthly payment, numberOfMonths, total payment, quote created date*/
        for (int i = 1; i < 180; i++)
            System.out.print("_");
        System.out.printf("\n|%-30s|%10s|%20s|%20s|%20s|%20s|%20s|%30s|\n", "Banks", "Codes", "AnnualInterestRate", "LoanAmount", "monthlyPayment", "numberOfYears", "totalPayment", "Date");
        for (int i = 1; i < 180; i++)
            System.out.print("_");

        ArrayList<Bank> matchedBanks = new ArrayList<>();

        /*find loan providers*/
        Arrays.stream(banks).filter(bank -> bank.getLoanInfo().getAnnualInterestRate() <= annualInterestRate)
                .forEach(bank -> matchedBanks.add(findMatch(bank, annualInterestRate, numberOfYears, loanAmount)));

        System.out.println();
        for (int i = 1; i < 180; i++)
            System.out.print("_");
        /*print total match found-->count of loan providers*/
        System.out.println("\n" + matchedBanks.size() + " Match found");
    }

    /* get each loan provider matches the criteria --> AnnualInterestRate less than or equal to user requested AnnualInterestRate */
    public Bank findMatch(Bank bank, double annualInterestRate, int numberOfYears, double loanAmount) {
        bank.getLoanInfo().setAnnualInterestRate(annualInterestRate)
                .setNumberOfYears(numberOfYears)
                .setLoanAmount(loanAmount);
        System.out.printf("\n|%-30s|%10s|%20s|%20s|%20s|%20s|%20s|%30s|",
                bank.getBankName(),
```



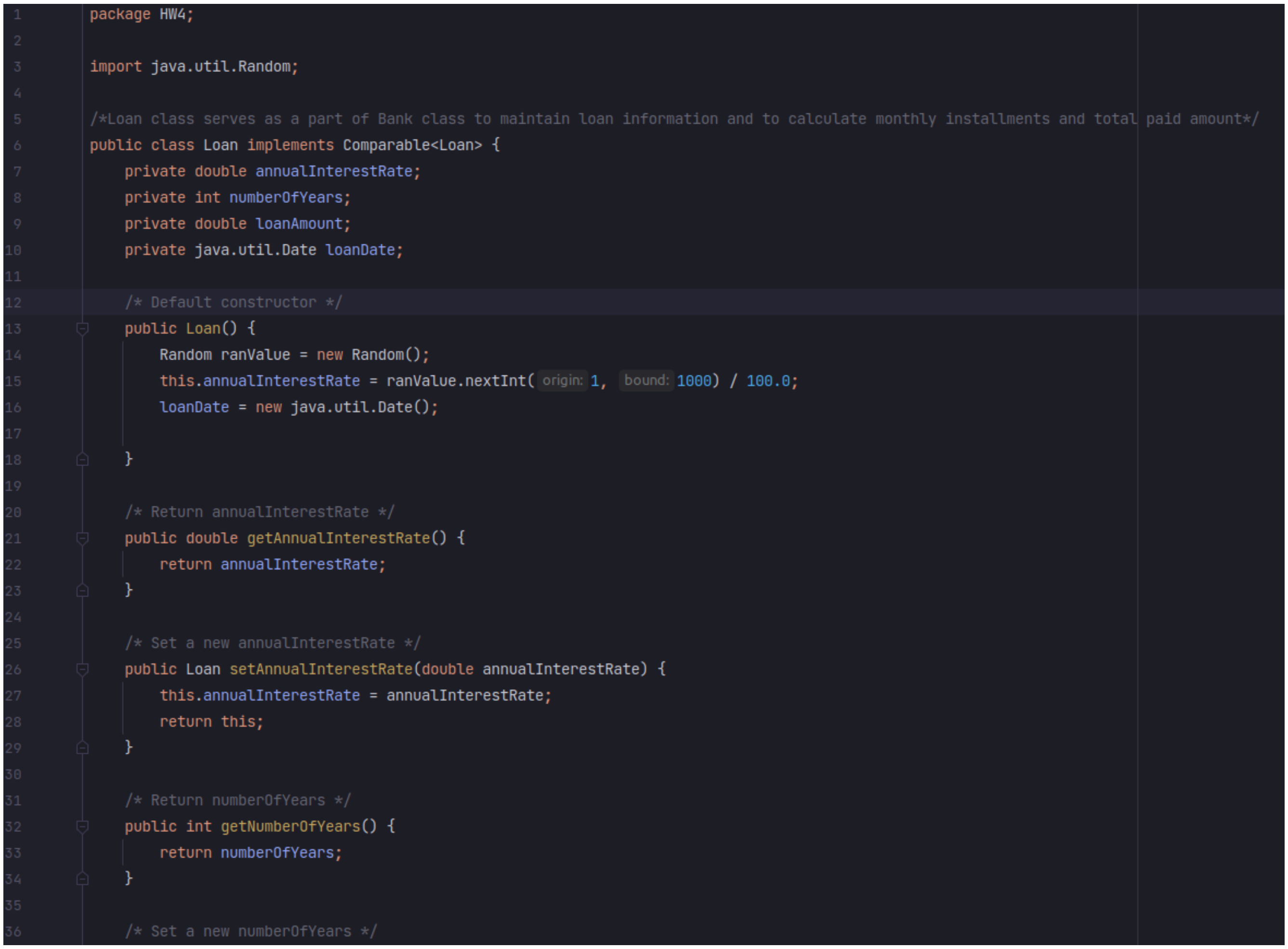
```java
                bank.getBankCode(),
                bank.getLoanInfo().getAnnualInterestRate(),
                bank.getLoanInfo().getLoanAmount(),
                bank.getLoanInfo().getMonthlyPayment(),
                bank.getLoanInfo().getNumberOfYears(),
                bank.getLoanInfo().getTotalPayment(),
                bank.getLoanInfo().getLoanDate());
        return bank;
    }

    public void sortBank(Bank[] b, SortType typeValue) {
        if (typeValue == SortType.NAME) {
            Arrays.stream(b).sorted(Comparator.comparing(Bank::getBankName).thenComparing(Bank::getLoanInfo)).toList().toArray(b);
        } else {
            Arrays.stream(b).sorted(Comparator.comparing(Bank::getLoanInfo).thenComparing(Bank::getBankName)).toList().toArray(b);
        }
    }

    private enum SortType {
        NAME,
        INTEREST_RATE
    }
}
```



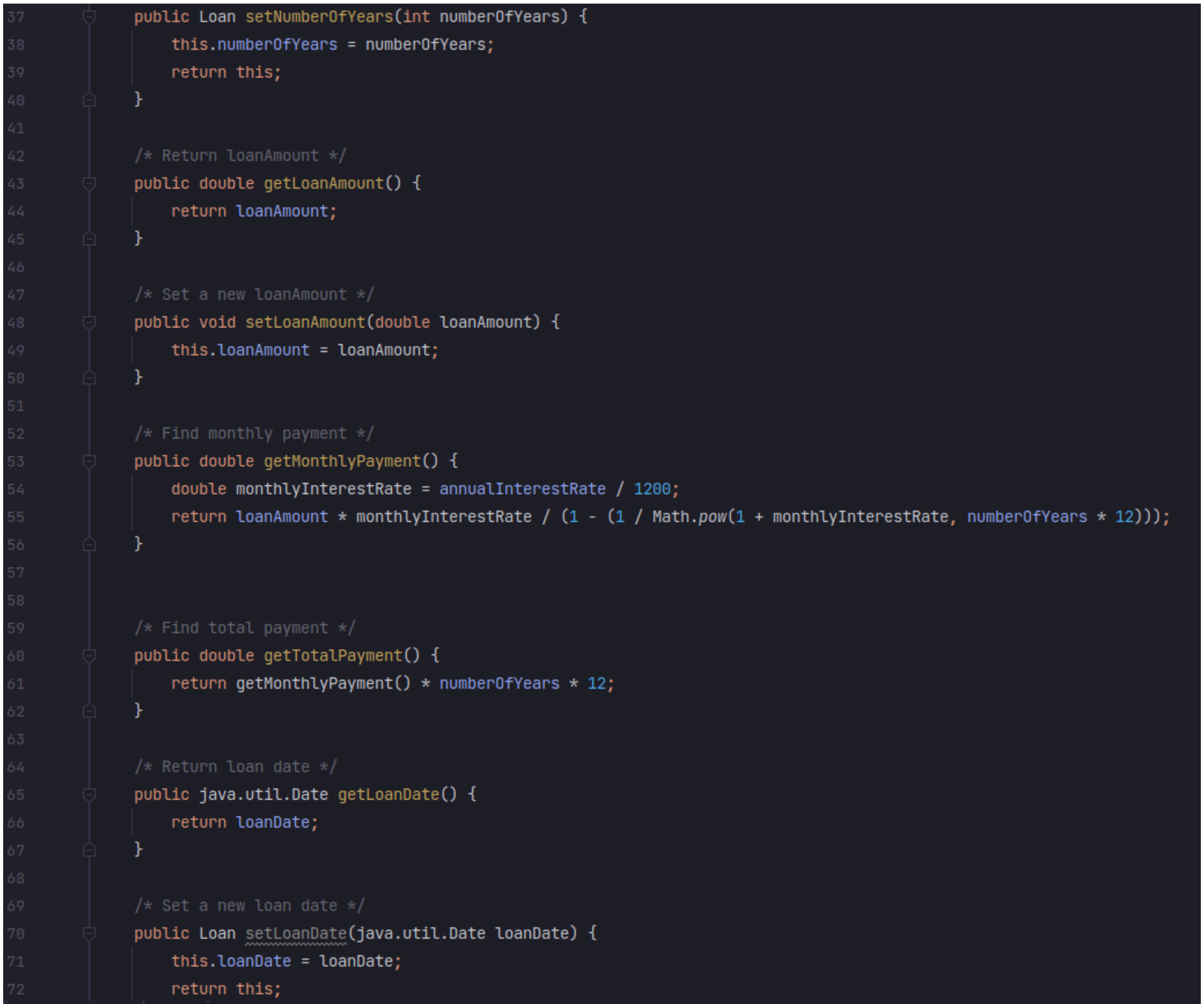
```java
package HW4;

import java.util.Random;

/*Bank class maintains banks information.*/
public class Bank {
    private Loan loanInfo;
    private String bankName;
    private int bankCode;

    /* constructor to create initial banks with names, codes, and loan information with annualInterestRate, and quote created date*/
    Bank() {
        Random ranValue = new Random();
        String[] bankNames = {"Wells Fargo", "City Bank", "Chase", "Discovery", "Terra Bank", "PNC Financial Services", "Capital One Financial"};
        this.setBankName(bankNames[ranValue.nextInt(bankNames.length)]);
        this.bankCode = 1 + ranValue.nextInt( bound: 99);
        loanInfo = new Loan();
    }

    /*prints all banks with bank name, bank codes, bank annualInterestRate, and quote created date */
    public void printBanksCode() {
        System.out.printf("\n|%-30s|%10s|%20s|%30s|", this.bankName, this.bankCode, this.getLoanInfo().getAnnualInterestRate(), this.getLoanInfo().getLoanDate());
    }

    /*returns bankName */
    public String getBankName() {
        return bankName;
    }

    /*sets a new bankName*/
    public void setBankName(String bankName) {
        this.bankName = bankName;
    }

    /*returns bankCode */
    public int getBankCode() {
```



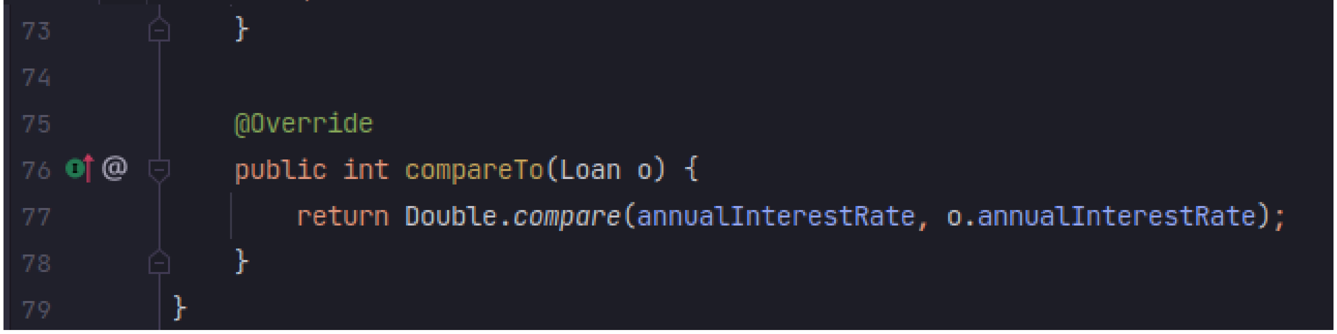
```java
        return bankCode;
    }

    /*sets a new bankCode*/
    public void setBankCode(int bankCode) {
        this.bankCode = bankCode;
    }

    /*returns loanInfo */
    public Loan getLoanInfo() {
        return loanInfo;
    }

    /*sets a new loanInfo*/
    public void setLoanInfo(Loan loanInfo) {
        this.loanInfo = loanInfo;
    }
}
```



```java
package HW4;

import java.util.Random;

/*Loan class serves as a part of Bank class to maintain loan information and to calculate monthly installments and total paid amount*/
public class Loan implements Comparable<Loan> {
    private double annualInterestRate;
    private int numberOfYears;
    private double loanAmount;
    private java.util.Date loanDate;

    /* Default constructor */
    public Loan() {
        Random ranValue = new Random();
        this.annualInterestRate = ranValue.nextInt( origin: 1,  bound: 1000) / 100.0;
        loanDate = new java.util.Date();

    }

    /* Return annualInterestRate */
    public double getAnnualInterestRate() {
        return annualInterestRate;
    }

    /* Set a new annualInterestRate */
    public Loan setAnnualInterestRate(double annualInterestRate) {
        this.annualInterestRate = annualInterestRate;
        return this;
    }

    /* Return numberOfYears */
    public int getNumberOfYears() {
        return numberOfYears;
    }

    /* Set a new numberOfYears */
```



```java
    public Loan setNumberOfYears(int numberOfYears) {
        this.numberOfYears = numberOfYears;
        return this;
    }

    /* Return loanAmount */
    public double getLoanAmount() {
        return loanAmount;
    }

    /* Set a new loanAmount */
    public void setLoanAmount(double loanAmount) {
        this.loanAmount = loanAmount;
    }

    /* Find monthly payment */
    public double getMonthlyPayment() {
        double monthlyInterestRate = annualInterestRate / 1200;
        return loanAmount * monthlyInterestRate / (1 - (1 / Math.pow(1 + monthlyInterestRate, numberOfYears * 12)));
    }


    /* Find total payment */
    public double getTotalPayment() {
        return getMonthlyPayment() * numberOfYears * 12;
    }

    /* Return loan date */
    public java.util.Date getLoanDate() {
        return loanDate;
    }

    /* Set a new loan date */
    public Loan setLoanDate(java.util.Date loanDate) {
        this.loanDate = loanDate;
        return this;
```



```java
    }

    @Override
    public int compareTo(Loan o) {
        return Double.compare(annualInterestRate, o.annualInterestRate);
    }
}
```