Rahul Nagaraju | **A20543969 |** [rnagaraju@hawk.iit.edu](mailto:rnagaraju@hawk.iit.edu)

**Lab2**

**Object-oriented App Development ( ITMD-510 )**

**PROJECT Bank record generations 100 points**

Objective: To write a program that parses and processes bank data from a file.

***PROJECT DESCRIPTION***

Bank of IIT has gotten their hands on some interesting data which will allow for possible loans to various clients from various regions.

Accompanying the labs specs is a csv (comma separated value) file named

**bank-Detail.csv** which contains valuable raw data to allow the bank to process loans based on client details from the file.

You need to parse the data and print record data for future loan considerations.

**Snapshot :-**

**A screenshot of a computer

Description automatically generated**

Source Code:

**Client.java**

    /\*

--------------------------------------------------------------------

- Author Rahul Nagaraju

- Assignment: Lab2

- FileName: Client.java

- Course: ITMD-510 Object-Oriented App Development

- Instructor: James Papademas

----------------------------------------------------------------------

\*/

public abstract class Client {

    public abstract void readData();

    public abstract void processData();

    public abstract void printData();

}

**BankRecords.java**

/\*

--------------------------------------------------------------------

- Author Rahul Nagaraju

- Assignment: Lab2

- FileName: BankRecords.java

- Course: ITMD-510 Object-Oriented App Development

- Instructor: James Papademas

----------------------------------------------------------------------

\*/

import java.io.BufferedReader; // Import the BufferedReader class for file reading

import java.io.FileReader; // Import the FileReader class for file reading

import java.io.File; // Import the File class for file manipulation

import java.io.FileNotFoundException; // Import the FileNotFoundException class for handling missing files

import java.io.IOException; // Import the IOException class for handling input/output errors

import java.util.ArrayList; // Import the ArrayList class for working with lists

import java.util.Arrays; // Import the Arrays class for working with arrays

import java.util.List; // Import the List class for working with lists

public class BankRecords extends Client{

    // Necessary variables

    private String id;

    private int age;

    private String sex;

    private String region;

    private double income;

    private String married;

    private int children;

    private String car;

    private String save\_act;

    private String current\_act;

    private String mortgage;

    private String pep;

    // Create a list to store lists of strings (2D list)

    List<List<String>> arrayOfLists = new ArrayList<>();

    // Create an array to hold BankRecords objects

    BankRecords[] recordObjects = new BankRecords[600];

    // Constructor

    public BankRecords() {

        // Default constructor

    }

    // Getter method for retrieving the ID of the bank record.

    public String getId() {

        return id;

    }

    // Setter method for setting the ID of the bank record.

    public void setId(String id) {

        this.id = id;

    }

    // Getter method for retrieving the age of the bank record.

    public int getAge() {

        return age;

    }

    // Setter method for setting the age of the bank record.

    public void setAge(int age) {

        this.age = age;

    }

    // Getter method for retrieving the sex of the bank record.

    public String getSex() {

        return sex;

    }

    // Setter method for setting the sex of the bank record.

    public void setSex(String sex) {

        this.sex = sex;

    }

    // Getter method for retrieving the region of the bank record.

    public String getRegion() {

        return region;

    }

    // Setter method for setting the region of the bank record.

    public void setRegion(String region) {

        this.region = region;

    }

    // Getter method for retrieving the income of the bank record.

    public double getIncome() {

        return income;

    }

    // Setter method for setting the income of the bank record.

    public void setIncome(double income) {

        this.income = income;

    }

    // Getter method for retrieving the marital status of the bank record.

    public String getMarried() {

        return married;

    }

    // Setter method for setting the marital status of the bank record.

    public void setMarried(String married) {

        this.married = married;

    }

    // Getter method for retrieving the number of children of the bank record.

    public int getChildren() {

        return children;

    }

    // Setter method for setting the number of children of the bank record.

    public void setChildren(int children) {

        this.children = children;

    }

    // Getter method for retrieving the car ownership status of the bank record.

    public String getCar() {

        return car;

    }

    // Setter method for setting the car ownership status of the bank record.

    public void setCar(String car) {

        this.car = car;

    }

    // Getter method for retrieving the savings account status of the bank record.

    public String getSave\_act() {

        return save\_act;

    }

    // Setter method for setting the savings account status of the bank record.

    public void setSave\_act(String save\_act) {

        this.save\_act = save\_act;

    }

    // Getter method for retrieving the current account status of the bank record.

    public String getCurrent\_act() {

        return current\_act;

    }

    // Setter method for setting the current account status of the bank record.

    public void setCurrent\_act(String current\_act) {

        this.current\_act = current\_act;

    }

    // Getter method for retrieving the mortgage status of the bank record.

    public String getMortgage() {

        return mortgage;

    }

    // Setter method for setting the mortgage status of the bank record.

    public void setMortgage(String mortgage) {

        this.mortgage = mortgage;

    }

    // Getter method for retrieving the PEP (Personal Equity Plan) status of the bank record.

    public String getPep() {

        return pep;

    }

    // Setter method for setting the PEP (Personal Equity Plan) status of the bank record.

    public void setPep(String pep) {

        this.pep = pep;

    }

    // Override the abstract methods from the abstract class by adding instance fields to read, process, and print the data in concrete subclasses.

        /\*\*

     \* Read data from a CSV file and store it in arrayOfLists.

     \*/

    @Override

    public void readData() {

        BufferedReader br = null; // Declare a BufferedReader variable

        try {

            // Initialize a reader object and set the file path to the project's root

            br = new BufferedReader(new FileReader(new File("bank-Detail.csv"))); // Create a FileReader and BufferedReader for reading the CSV file

            String line; // Declare a variable to store each line of the file

            // Read each record in the CSV file

            while ((line = br.readLine()) != null) { // Read each line of the file until the end

                // Split each record in the CSV file by a comma (,) and add it to the list

                List<String> parts = Arrays.asList(line.split(",")); // Split the line into a list of strings using a comma as the separator

                arrayOfLists.add(parts); // Add the list of strings to arrayOfLists

            }

        } catch (FileNotFoundException e) { // Handle the FileNotFoundException

            System.err.println("File not found: " + e.getMessage()); // Print an error message if the file is not found

        } catch (IOException e) { // Handle the IOException

            System.err.println("Error reading the file: " + e.getMessage()); // Print an error message if there is an IO error

        } finally {

            try {

                if (br != null) {

                    br.close(); // Close the BufferedReader if it is not null

                }

            } catch (IOException e) {

                System.err.println("Error closing the file: " + e.getMessage()); // Print an error message if there is an error while closing the file

            }

        }

        // Call the processData method to further process the data

        processData();

    }

    /\*\*

     \* Process the data from arrayOfLists and populate the recordObjects array.

     \*/

    @Override

    public void processData(){

       // Iterate through the data and populate recordObjects array

       int idx = 0; // Initialize an index variable

       for (List<String> rowData : arrayOfLists) { // Loop through each list of strings in arrayOfLists

            recordObjects[idx] = new BankRecords(); // Create a new BankRecords object

            recordObjects[idx].setId(rowData.get(0)); // Get the 1st column and set it as the ID in the BankRecords object

            recordObjects[idx].setAge(Integer.parseInt(rowData.get(1))); // Get the 2nd column, parse it as an integer, and set it as the age in the BankRecords object

            recordObjects[idx].setSex(rowData.get(2)); // Get the 3rd column and set it as the sex in the BankRecords object

            recordObjects[idx].setRegion(rowData.get(3)); // Get the 4th column and set it as the region in the BankRecords object

            recordObjects[idx].setIncome(Double.parseDouble(rowData.get(4))); // Get the 5th column, parse it as a double, and set it as the income in the BankRecords object

            recordObjects[idx].setMarried(rowData.get(5)); // Get the 6th column and set it as the marital status in the BankRecords object

            recordObjects[idx].setChildren(Integer.parseInt(rowData.get(6))); // Get the 7th column, parse it as an integer, and set it as the number of children in the BankRecords object

            recordObjects[idx].setCar(rowData.get(7)); // Get the 8th column and set it as the car ownership status in the BankRecords object

            recordObjects[idx].setSave\_act(rowData.get(8)); // Get the 9th column and set it as the savings account status in the BankRecords object

            recordObjects[idx].setCurrent\_act(rowData.get(9)); // Get the 10th column and set it as the current account status in the BankRecords object

            recordObjects[idx].setMortgage(rowData.get(10)); // Get the 11th column and set it as the mortgage status in the BankRecords object

            recordObjects[idx].setPep(rowData.get(11)); // Get the 12th column and set it as the PEP (Personal Equity Plan) status in the BankRecords object

            // Increment the index for the next BankRecords object

            idx++;

       }

       // Call the printData method to print the data

       printData();

    }

    /\*\*

     \* Print the data in a formatted tabular format.

     \*/

    @Override

    public void printData() {

        // Print column headers

        System.out.printf("%-10s %-4s %-6s %-10s %-10s %-10s %-8s %-4s %-6s %-8s %-8s %-3s%n",

                "ID", "Age", "Sex", "Region", "Income", "Married", "Children", "Car", "Savings", "Current", "Mortgage", "PEP");

        // Print the first 25 records (or fewer if there are fewer than 25 records)

        int maxRecords = Math.min(25, recordObjects.length);

        // Print records

        for (int i = 0; i < maxRecords; i++) { // Loop through each BankRecords object in recordObjects

            BankRecords record = recordObjects[i];

            // Print the data in a formatted tabular format

            System.out.printf("%-10s %-4d %-6s %-10s %-10.2f %-10s %-8d %-4s %-6s %-8s %-8s %-3s%n",

                    record.getId(), record.getAge(), record.getSex(), record.getRegion(), record.getIncome(),

                    record.getMarried(), record.getChildren(), record.getCar(), record.getSave\_act(),

                    record.getCurrent\_act(), record.getMortgage(), record.getPep());

        }

    }

}

**BankRecordsTest.java**

/\*

--------------------------------------------------------------------

- Author Rahul Nagaraju

- Assignment: Lab2

- FileName: BankRecordsTest.java

- Course: ITMD-510 Object-Oriented App Development

- Instructor: James Papademas

----------------------------------------------------------------------

\*/

public class BankRecordsTest {

    public static void main(String[] args) {

        // Create an instance of the BankRecords class

        BankRecords bankRecords = new BankRecords();

        // Call the readData() method to read data from a CSV file and process it

        bankRecords.readData();

    }

}