**Lab 3 – Bank Records Generations**

By: Laura Pereda

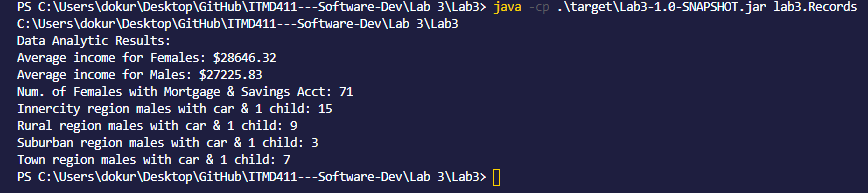
Note:

To run jar file:

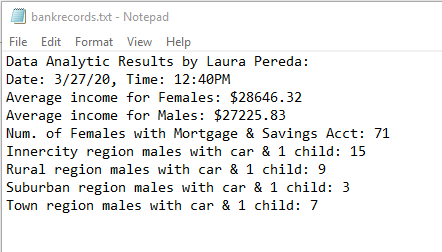


Snapshots:

* Console:



* Bankrecorsds.txt file:



Source Code:

* Records.java
* package lab3;
* import java.io.FileWriter;
* import java.io.IOException;
* import java.text.DecimalFormat;
* import java.util.ArrayList;
* import java.util.Collections;
* public class Records extends BankRecords {
* static FileWriter fw = null;
* public Records() {
* try{
* fw = new FileWriter("bankrecords.txt");
* } catch (IOException e) {
* e.printStackTrace();
* }
* }
* */\**
* *\* Constructor with parameters*
* *\*/*
* public Records(String id, int age, String sex, String region, double income, String martialStatus,
* int totalChildren, String hasCar, String save\_act, String current\_act, String mortgage, String pep) {
* super(id, age, sex, region, income, martialStatus, totalChildren, hasCar, save\_act, current\_act, mortgage, pep);
* }
* public static void main(String[] args) {
* Records records = new Records();
* records.readData();
* *// Calculate the average income for sex*
* System.out.println("Data Analytic Results: ");
* AvgIncome(records.getbArrayList());
* locationComp(records.getbArrayList());
* *// Close out file object*
* try {
* fw.close();
* } catch (IOException e) {
* e.printStackTrace();
* }
* }
* */\**
* *\* Access bArrayList in BankRecords*
* *\*/*
* @Override
* public ArrayList<BankRecords> getbArrayList() {
* return super.getbArrayList();
* }
* */\**
* *\* Method to determine average income by sec*
* *\* Also counts females with mortgage & sav. acct*
* *\*/*
* public static void AvgIncome(ArrayList<BankRecords> arrayList){
* *// Sort the data by males vs females...*
* Collections.sort(arrayList, new SexComparator());
* */\**
* *\*   To keep track of:*
* *\*   Counts for # of males & females*
* *\*   Total income from males & females*
* *\*/*
* int mCount, fCount, mort\_sav;
* double totalMIncome, totalFIncome;
* mCount = fCount = mort\_sav = 0;
* totalFIncome = totalMIncome = 0;
* for (int i = 0; i < arrayList.size(); i++){
* if (arrayList.get(i).getSex().equals("FEMALE")){
* *// Increment fCount*
* fCount++;
* *// Add income to totalFIncome*
* totalFIncome += arrayList.get(i).getIncome();
* if(arrayList.get(i).getMortgage().equals("YES") && arrayList.get(i).getSave\_act().equals("YES")){
* mort\_sav++;
* }
* } else {
* *// Increment mCount*
* mCount++;
* *// Add income to totalMIncome*
* totalMIncome += arrayList.get(i).getIncome();
* }
* }
* double avgFIncome = totalFIncome/fCount;
* double avgMIncome = totalMIncome/mCount;
* *// Try to write to file obj*
* DecimalFormat df = new DecimalFormat("##.##");
* try {
* fw.write("Data Analytic Results by Laura Pereda: " + "\n");
* fw.write("Date: 3/27/20, Time: 12:40PM");
* fw.write("Average income for Females: $" + df.format(avgFIncome) + "\n");
* fw.write("Average income for Males: $" + df.format(avgMIncome)  + "\n");
* fw.write("Num. of Females with Mortgage & Savings Acct: " + mort\_sav  + "\n");
* } catch (IOException e) {
* e.printStackTrace();
* }
* System.out.println("Average income for Females: $" + df.format(avgFIncome));
* System.out.println("Average income for Males: $" + df.format(avgMIncome));
* System.out.println("Num. of Females with Mortgage & Savings Acct: " + mort\_sav);
* }
* public static void locationComp(ArrayList<BankRecords> arrayList) {
* *// Sort the records based on Location*
* Collections.sort(arrayList, new LocationComparator());
* */\**
* *\*   Keep track of children per region*
* *\*/*
* int innerChildren, ruralChildren, suburbanChildren, townChildren;
* innerChildren = ruralChildren = suburbanChildren = townChildren = 0;
* for (int i = 0; i < arrayList.size(); i++){
* if (arrayList.get(i).getHasCar().equals("YES") && arrayList.get(i).getTotalChildren() == 1 && arrayList.get(i).getSex().equals("MALE")) {
* if (arrayList.get(i).getRegion().equals("INNER\_CITY")) {
* innerChildren++;
* } else if (arrayList.get(i).getRegion().equals("RURAL")) {
* ruralChildren++;
* } else if (arrayList.get(i).getRegion().equals("SUBURBAN")) {
* suburbanChildren++;
* } else {
* townChildren++;
* }
* }
* }
* *// Write to exisiting file obj*
* try {
* fw.write("Innercity region males with car & 1 child: " + innerChildren  + "\n");
* fw.write("Rural region males with car & 1 child: " + ruralChildren  + "\n");
* fw.write("Suburban region males with car & 1 child: " + suburbanChildren  + "\n");
* fw.write("Town region males with car & 1 child: " + townChildren  + "\n");
* } catch (IOException e) {
* e.printStackTrace();
* }
* System.out.println("Innercity region males with car & 1 child: " + innerChildren);
* System.out.println("Rural region males with car & 1 child: " + ruralChildren);
* System.out.println("Suburban region males with car & 1 child: " + suburbanChildren);
* System.out.println("Town region males with car & 1 child: " + townChildren);
* }
* }
* LocationComparator.java
* package lab3;
* import java.util.Comparator;
* public class LocationComparator implements Comparator<BankRecords> {
* @Override
* public int compare(BankRecords o1, BankRecords o2) {
* int result = o1.getRegion().compareTo(o2.getRegion());
* return result;
* }
* }
* SexComparator.java

package lab3;

import java.util.Comparator;

public class SexComparator implements Comparator<BankRecords> {

    @Override

    public int compare(BankRecords o1, BankRecords o2) {

        int result = o1.getSex().compareTo(o2.getSex());

        return result;

    }

}

Lab 2 Code (In-case):

* Client.java
* package lab3;
* public abstract class Client{
* */\*\**
* *\*  Read from csv file*
* *\*/*
* public abstract void readData();
* */\*\**
* *\*  Print information from csv*
* *\*/*
* public abstract void printData();
* }
* BankRecords.java

package lab3;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.ArrayList;

*/\*\**

*\* Hello world!*

*\*/*

public class BankRecords extends Client {

*//BankRecords variables*

    private String id;

    private int age;

    private String sex;

    private String region;

    private double income;

    private String martialStatus;

    private int totalChildren;

    private String hasCar;

    private String save\_act;

    private String current\_act;

    private String mortgage;

    private String pep;

    private static ArrayList<BankRecords> bArrayList;

*/\**

*\*   Constructor with no parameters*

*\*/*

    protected BankRecords() {

    }

*/\**

*\*   Constructor with parameters*

*\*/*

    public BankRecords(String id, int age, String sex, String region, double income, String martialStatus,

            int totalChildren, String hasCar, String save\_act, String current\_act, String mortgage, String pep) {

        this.id = id;

        this.age = age;

        this.sex = sex;

        this.region = region;

        this.income = income;

        this.martialStatus = martialStatus;

        this.totalChildren = totalChildren;

        this.hasCar = hasCar;

        this.save\_act = save\_act;

        this.current\_act = current\_act;

        this.mortgage = mortgage;

        this.pep = pep;

    }

*/\*\**

*\* Says hello to the world.*

*\* @param args The arguments of the program.*

*\*/*

    public static void main(String[] args) {

        BankRecords br = new BankRecords();

        br.readData();

        br.printData();

    }

    @Override

    public void readData() {

        BufferedReader br = null;

        String line = "";

        bArrayList = new ArrayList<>();

        try{

*//Create and initialize a new reader object to read from the CSV file*

            String workingDir = System.getProperty("user.dir");

            System.out.println(workingDir);

            FileReader fr = new FileReader(workingDir+"\\bank-Detail.csv");

            br = new BufferedReader(fr);

            while((line = br.readLine()) != null){

                String[] data = line.split(",");

*//Create an empty BankRecord object & set the data*

                BankRecords bankRecords = new BankRecords();

                bankRecords.setId(data[0]);                                         *//data[0] : String id*

                bankRecords.setAge(Integer.parseInt(data[1]));                      *//data[1] : int age*

                bankRecords.setSex(data[2]);                                        *//data[2] : String sex*

                bankRecords.setRegion(data[3]);                                     *//data[3] : String region*

                bankRecords.setIncome(Double.parseDouble(data[4]));                   *//data[4] : int income*

                bankRecords.setMartialStatus(data[5]);                              *//data[5] : String region*

                bankRecords.setTotalChildren(Integer.parseInt(data[6]));             *//data[6] : int children*

                bankRecords.setHasCar(data[7]);                                     *//data[7] : String car*

                bankRecords.setSave\_act(data[8]);                                   *//data[8] : String sav\_act*

                bankRecords.setCurrent\_act(data[9]);                                *//data[9] : String curr\_act*

                bankRecords.setMortgage(data[10]);                                  *//data[10]: String mortgage*

                bankRecords.setPep(data[11]);                                       *//data[11]: String pep*

*//Add object to empty bArrayList*

                bArrayList.add(bankRecords);

            }

*//Call printData()*

*// processData();*

        } catch (FileNotFoundException e){

            e.printStackTrace();

        } catch (IOException e){

            e.printStackTrace();

        } finally {

            if (br != null){

                try{

                    br.close();

                } catch (IOException e){

                    e.printStackTrace();

                }

            }

        }

    }

    @Override

    public void printData() {

        System.out.println("Programmer: Laura Pereda");

*//Header to print information*

        System.out.println("ID\t\tAGE\t\tSEX\t\tREGION\t\tINCOME\t\t\tMORTGAGE");

*//Created a for loop to only print out the first 25 objects with information*

        for(int i = 0; i <= 25; i++){

            String id = bArrayList.get(i).getId();

            String removeID = id.replace("id", "");

            int age = bArrayList.get(i).getAge();

            String sex = bArrayList.get(i).getSex();

            String region = bArrayList.get(i).getRegion();

            Double income = bArrayList.get(i).getIncome();

            String mortgage = bArrayList.get(i).getMortgage();

*/\*  Because of formatting issues, I decided to abbreiviate REGION*

*\*   INNER\_CITY = IC*

*\*   SUBURBAN = SUB*

*\*   Note: I removed the string "id" from the ID of each object since it is redundant to maintain*

*\*/*

            String abbriviatedRegion = "";

            String printOut = "";

            if(region.equals("INNER\_CITY")){

                abbriviatedRegion = "IC";

                printOut = String.format("%s\t\t%d\t\t%s\t\t%s\t\t%.3f\t\t%s", removeID, age, sex, abbriviatedRegion, income, mortgage);

            } else if(region.equals("SUBURBAN")) {

                abbriviatedRegion = "SUB";

                printOut = String.format("%s\t\t%d\t\t%s\t\t%s\t\t%.3f\t\t%s", removeID, age, sex, abbriviatedRegion, income, mortgage);

            } else {

                printOut = String.format("%s\t\t%d\t\t%s\t\t%s\t\t%.3f\t\t%s", removeID, age, sex, region, income, mortgage);

            }

            System.out.println(printOut);

*//System.out.println(id + "\t\t" + age + "\t\t" + sex + "\t\t" + region + "\t\t" + income + "\t\t" + mortgage);*

        }

    }

    public String getId() {

        return id;

    }

    public void setId(String id) {

        this.id = id;

    }

    public int getAge() {

        return age;

    }

    public void setAge(int age) {

        this.age = age;

    }

    public String getSex() {

        return sex;

    }

    public void setSex(String sex) {

        this.sex = sex;

    }

    public String getRegion() {

        return region;

    }

    public void setRegion(String region) {

        this.region = region;

    }

    public double getIncome() {

        return income;

    }

    public void setIncome(double income) {

        this.income = income;

    }

    public String getMartialStatus() {

        return martialStatus;

    }

    public void setMartialStatus(String martialStatus) {

        this.martialStatus = martialStatus;

    }

    public int getTotalChildren() {

        return totalChildren;

    }

    public void setTotalChildren(int totalChildren) {

        this.totalChildren = totalChildren;

    }

    public String getHasCar() {

        return hasCar;

    }

    public void setHasCar(String hasCar) {

        this.hasCar = hasCar;

    }

    public String getSave\_act() {

        return save\_act;

    }

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

        this.save\_act = save\_act;

    }

    public String getCurrent\_act() {

        return current\_act;

    }

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

        this.current\_act = current\_act;

    }

    public String getMortgage() {

        return mortgage;

    }

    public void setMortgage(String mortgage) {

        this.mortgage = mortgage;

    }

    public String getPep() {

        return pep;

    }

    public void setPep(String pep) {

        this.pep = pep;

    }

    public ArrayList<BankRecords> getbArrayList() {

        return bArrayList;

    }

    public static void setbArrayList(ArrayList<BankRecords> bArrayList) {

        BankRecords.bArrayList = bArrayList;

    }

    @Override

    public String toString() {

        return "BankRecords [age=" + age + ", current\_act=" + current\_act + ", hasCar=" + hasCar + ", id=" + id

                + ", income=" + income + ", martialStatus=" + martialStatus + ", mortgage=" + mortgage + ", pep=" + pep

                + ", region=" + region + ", save\_act=" + save\_act + ", sex=" + sex + ", totalChildren=" + totalChildren

                + "]";

    }

}