**Records.java**

**import** java.io.FileWriter;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Arrays;

**import** java.text.DecimalFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Calendar;

**public** **class** Records **extends** BankRecords

{

**static** FileWriter *fw* = **null**;

File file = **new** File("bankrecords.txt");

DecimalFormat df = **new** DecimalFormat("0.00");

**public** Records()

{

**try** {

*fw* = **new** FileWriter(file);

}

**catch**(IOException e1)

{

e1.printStackTrace();

}

}

**public** **static** **void** main(String[] args)

{

Records br = **new** Records();

br.readData();

*AverageComp*();

*MortgageASavings*();

*CarAChild*();

**try** {

*fw*.close();

}

**catch**(IOException e2)

{

e2.printStackTrace();

}

}

**private** **static** **void** AverageComp()

{

Arrays.*sort*(*robjs*, **new** GenderComparator());

DecimalFormat df = **new** DecimalFormat("0.00");

**double** fsum = 0;

**double** msum = 0;

**double** fCt = 0;

**double** mCt = 0;

**for**(**int** i = 0; i < *robjs*.length; i++)

{

**if**(*robjs*[i].getSex().equals("FEMALE"))

{

fsum += *robjs*[i].getIncome();

++fCt;

}

**else**

{

msum += *robjs*[i].getIncome();

++mCt;

}

}

//print resulting averages to console and txt file

**double** femAvg;

**double** maleAvg;

femAvg = fsum/(fCt);

maleAvg = msum/(mCt);

System.***out***.println("Calculating averages...");

**try** {

*fw*.write("Averages\n================\n-Income for females: $" + df.format(femAvg));

*fw*.write("...\n");

*fw*.write("-Income for male: $" + df.format(maleAvg));

*fw*.write("...\n");

System.***out***.println("Averages\n================\n");

System.***out***.println("Income for females: $" + df.format(femAvg));

System.***out***.println("Income for male: $" + df.format(maleAvg));

}

**catch**(IOException e3)

{

e3.printStackTrace();

}

}

**private** **static** **void** MortgageASavings()

{

GenderComparator gc = **new** GenderComparator();

Arrays.*sort*(*robjs*, gc);

**int** count = 0;

**for**(**int** i = 0; i < *robjs*.length; i++)

{

**if**(*robjs*[i].getSex().equals("FEMALE"))

{

**if**(*robjs*[i].getMortgage().equals("YES") && *robjs*[i].getSave\_act().equals("YES"))

{

//If the person if female and has

//both a mortgage and savings account

//increment the counter

++count;

}

}

}

System.***out***.println("Calculating...");

**try** {

DecimalFormat df = **new** DecimalFormat("0.00");

*fw*.write("\nFemales w/ Mortgage and Savings Account: " + count);

*fw*.write("...");

System.***out***.println("\nFemales w/ Mortgage and Savings Account: " + count);

}

**catch**(IOException e3)

{

e3.printStackTrace();

}

}

**private** **static** **void** CarAChild()

{

SimpleComparator sc = **new** SimpleComparator();

Arrays.*sort*(*robjs*, sc);

**int** ICcount = 0;

**int** Tcount = 0;

**int** Rcount = 0;

**int** Scount = 0;

**int** count = 0;

**for**(**int** i = 0; i < *robjs*.length; i++)

{

**if**(*robjs*[i].getSex().equals("MALE"))

{

**if**(*robjs*[i].getCar().equals("YES") && *robjs*[i].getChildren() == 1)

{

**switch**(*robjs*[i].getRegion())

{

**case** "INNER\_CITY":

ICcount++;

**break**;

**case** "TOWN":

Tcount++;

**break**;

**case** "RURAL":

Rcount++;

**break**;

**case** "SUBURBAN":

Scount++;

**break**;

**default**:

count++;

**break**;

}

}

}

}

**try** {

String timeStamp = **new** SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(Calendar.*getInstance*().getTime());

*fw*.write(System.*lineSeparator*());

*fw*.write("\nInnercity region males with car & 1 child: " + ICcount);

*fw*.write("\nRural region males with car & 1 child: " + Rcount);

*fw*.write("\nSuburban region males with car & 1 child: " + Scount);

*fw*.write("\nTown region males with car & 1 child: " + Tcount);

*fw*.write("...");

*fw*.write("Current Time Stamp = " + timeStamp + "\nProgrammed by Carlos Lopez\n");

System.***out***.println("\nInnercity region males with car & 1 child: " + ICcount);

System.***out***.println("Rural region males with car & 1 child: " + Rcount);

System.***out***.println("Suburban region males with car & 1 child: " + Scount);

System.***out***.println("Town region males with car & 1 child: " + Tcount);

System.***out***.println("\nCurrent Time Stamp = " + timeStamp + "\nProgrammed by Carlos Lopez\n");

}

**catch**(IOException e3)

{

e3.printStackTrace();

}

}

}

**BankRecords.java**

/\*

Use

\*/

import java.util.\*;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.File;

import java.io.\*;

public class BankRecords extends Client

{

String id;

String sex; // Male or female

String region; // INNER\_CITY, TOWN, RURAL, SUBURBAN

String married; // YES, NO

String car; //YES, NO

String save\_act; //YES, NO

String current\_act; //YES, NO

String mortgage; //YES, NO

String pep; //YES, NO

int age;

int children; // 0, 1, 2, 3

double income;

static BankRecords robjs[] = new BankRecords[600];

static ArrayList<List<String>> array = new ArrayList<>();

public BankRecords()

{

id = "None";

sex = "Male";

region = "INNER\_CITY";

married = "YES";

car = "YES";

save\_act = "YES";

current\_act = "YES";

mortgage = "YES";

pep = "YES";

age = 22;

children = 1;

income = 25000.00;

}

@Override

public void readData()

{

// TODO Auto-generated method stub

System.out.println("Read Data");

BufferedReader br;

FileReader fr;

File file;

String line;

try

{

file = new File("bank-Detail.csv");

fr = new FileReader(file);

br = new BufferedReader(fr);

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

{

array.add(Arrays.asList(line.split(",")));

}

}

catch(IOException e1)

{

e1.printStackTrace();

System.out.println("File Not Found...");

}

processData();

}

@Override

public void processData()

{

try{

// TODO Auto-generated method stub

System.out.println("Process Data");

int indx = 0;

for(List<String> rowData: array)

{

robjs[indx] = new BankRecords();

robjs[indx].setId(rowData.get(0));

robjs[indx].setAge(Integer.parseInt(rowData.get(1)));

robjs[indx].setSex(rowData.get(2));

robjs[indx].setRegion(rowData.get(3));

robjs[indx].setIncome(Double.parseDouble(rowData.get(4)));

robjs[indx].setMarried(rowData.get(5));

robjs[indx].setChildren(Integer.parseInt(rowData.get(6)));

robjs[indx].setCar(rowData.get(7));

robjs[indx].setSave\_act(rowData.get(8));

robjs[indx].setCurrent\_act(rowData.get(9));

robjs[indx].setMortgage(rowData.get(10));

robjs[indx].setPep(rowData.get(11));

indx++;

}

printData();

}

catch(java.lang.NumberFormatException e2)

{

e2.printStackTrace();

}

}

@Override

public void printData()

{

// TODO Auto-generated method stub

System.out.println("Print Data");

System.out.println("\nID \tAge \tSex \tRegion \t\tIncome \tMarried? \tChildren \tCar? \tSavings Account? \tCurrent Account? \tMortgage? \tPEP?");

int indx = 0;

for(indx = 0; indx < 25; indx++)

{

System.out.print(robjs[indx].getId() + "\t");

System.out.print(robjs[indx].getAge() + "\t");

System.out.print(robjs[indx].getSex() + "\t");

System.out.print(robjs[indx].getRegion() + "\t");

if(robjs[indx].getRegion().equals("RURAL") || robjs[indx].getRegion().equals("TOWN"))

{

System.out.print("\t");

}

System.out.print(robjs[indx].getIncome() + "\t");

System.out.print(robjs[indx].getMarried() + "\t\t");

System.out.print(robjs[indx].getChildren() + "\t\t");

System.out.print(robjs[indx].getCar() + "\t");

System.out.print(robjs[indx].getSave\_act() + "\t\t\t");

System.out.print(robjs[indx].getCurrent\_act() + "\t\t\t");

System.out.print(robjs[indx].getMortgage() + "\t\t");

System.out.print(robjs[indx].getPep() + "\t\t\n");

indx++;

}

}

public String getId() {

return id;

}

public String getSex() {

return sex;

}

public String getRegion() {

return region;

}

public String getMarried() {

return married;

}

public String getCar() {

return car;

}

public String getSave\_act() {

return save\_act;

}

public String getCurrent\_act() {

return current\_act;

}

public String getMortgage() {

return mortgage;

}

public String getPep() {

return pep;

}

public int getAge() {

return age;

}

public int getChildren() {

return children;

}

public double getIncome() {

return income;

}

public void setId(String id) {

this.id = id;

}

public void setSex(String sex) {

this.sex = sex;

}

public void setRegion(String region) {

this.region = region;

}

public void setMarried(String married) {

this.married = married;

}

public void setCar(String car) {

this.car = car;

}

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

this.save\_act = save\_act;

}

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

this.current\_act = current\_act;

}

public void setMortgage(String mortgage) {

this.mortgage = mortgage;

}

public void setPep(String pep) {

this.pep = pep;

}

public void setAge(int age) {

this.age = age;

}

public void setChildren(int children) {

this.children = children;

}

public void setIncome(double income) {

this.income = income;

}

}

**BankRecords.java**

/\*

Use

\*/

import java.util.\*;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.File;

import java.io.\*;

public class BankRecords extends Client

{

String id;

String sex; // Male or female

String region; // INNER\_CITY, TOWN, RURAL, SUBURBAN

String married; // YES, NO

String car; //YES, NO

String save\_act; //YES, NO

String current\_act; //YES, NO

String mortgage; //YES, NO

String pep; //YES, NO

int age;

int children; // 0, 1, 2, 3

double income;

static BankRecords robjs[] = new BankRecords[600];

static ArrayList<List<String>> array = new ArrayList<>();

public BankRecords()

{

id = "None";

sex = "Male";

region = "INNER\_CITY";

married = "YES";

car = "YES";

save\_act = "YES";

current\_act = "YES";

mortgage = "YES";

pep = "YES";

age = 22;

children = 1;

income = 25000.00;

}

@Override

public void readData()

{

// TODO Auto-generated method stub

System.out.println("Read Data");

BufferedReader br;

FileReader fr;

File file;

String line;

try

{

file = new File("bank-Detail.csv");

fr = new FileReader(file);

br = new BufferedReader(fr);

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

{

array.add(Arrays.asList(line.split(",")));

}

}

catch(IOException e1)

{

e1.printStackTrace();

System.out.println("File Not Found...");

}

processData();

}

@Override

public void processData()

{

try{

// TODO Auto-generated method stub

System.out.println("Process Data");

int indx = 0;

for(List<String> rowData: array)

{

robjs[indx] = new BankRecords();

robjs[indx].setId(rowData.get(0));

robjs[indx].setAge(Integer.parseInt(rowData.get(1)));

robjs[indx].setSex(rowData.get(2));

robjs[indx].setRegion(rowData.get(3));

robjs[indx].setIncome(Double.parseDouble(rowData.get(4)));

robjs[indx].setMarried(rowData.get(5));

robjs[indx].setChildren(Integer.parseInt(rowData.get(6)));

robjs[indx].setCar(rowData.get(7));

robjs[indx].setSave\_act(rowData.get(8));

robjs[indx].setCurrent\_act(rowData.get(9));

robjs[indx].setMortgage(rowData.get(10));

robjs[indx].setPep(rowData.get(11));

indx++;

}

printData();

}

catch(java.lang.NumberFormatException e2)

{

e2.printStackTrace();

}

}

@Override

public void printData()

{

// TODO Auto-generated method stub

System.out.println("Print Data");

System.out.println("\nID \tAge \tSex \tRegion \t\tIncome \tMarried? \tChildren \tCar? \tSavings Account? \tCurrent Account? \tMortgage? \tPEP?");

int indx = 0;

for(indx = 0; indx < 25; indx++)

{

System.out.print(robjs[indx].getId() + "\t");

System.out.print(robjs[indx].getAge() + "\t");

System.out.print(robjs[indx].getSex() + "\t");

System.out.print(robjs[indx].getRegion() + "\t");

if(robjs[indx].getRegion().equals("RURAL") || robjs[indx].getRegion().equals("TOWN"))

{

System.out.print("\t");

}

System.out.print(robjs[indx].getIncome() + "\t");

System.out.print(robjs[indx].getMarried() + "\t\t");

System.out.print(robjs[indx].getChildren() + "\t\t");

System.out.print(robjs[indx].getCar() + "\t");

System.out.print(robjs[indx].getSave\_act() + "\t\t\t");

System.out.print(robjs[indx].getCurrent\_act() + "\t\t\t");

System.out.print(robjs[indx].getMortgage() + "\t\t");

System.out.print(robjs[indx].getPep() + "\t\t\n");

indx++;

}

}

public String getId() {

return id;

}

public String getSex() {

return sex;

}

public String getRegion() {

return region;

}

public String getMarried() {

return married;

}

public String getCar() {

return car;

}

public String getSave\_act() {

return save\_act;

}

public String getCurrent\_act() {

return current\_act;

}

public String getMortgage() {

return mortgage;

}

public String getPep() {

return pep;

}

public int getAge() {

return age;

}

public int getChildren() {

return children;

}

public double getIncome() {

return income;

}

public void setId(String id) {

this.id = id;

}

public void setSex(String sex) {

this.sex = sex;

}

public void setRegion(String region) {

this.region = region;

}

public void setMarried(String married) {

this.married = married;

}

public void setCar(String car) {

this.car = car;

}

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

this.save\_act = save\_act;

}

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

this.current\_act = current\_act;

}

public void setMortgage(String mortgage) {

this.mortgage = mortgage;

}

public void setPep(String pep) {

this.pep = pep;

}

public void setAge(int age) {

this.age = age;

}

public void setChildren(int children) {

this.children = children;

}

public void setIncome(double income) {

this.income = income;

}

}

**Bankrecords.txt**

Averages

================

-Income for females: $28646.32...

-Income for male: $27225.83...

Females w/ Mortgage and Savings Account: 71...

Innercity region males with car & 1 child: 15

Rural region males with car & 1 child: 9

Suburban region males with car & 1 child: 3

Town region males with car & 1 child: 7...Current Time Stamp = 2018/03/06 22:07:35

Programmed by Carlos Lopez

**SimpleComparator.java**

**import** java.util.Comparator;

**public** **class** SimpleComparator **implements** Comparator<BankRecords> {

@Override

**public** **int** compare(BankRecords arg0, BankRecords arg1) {

// **TODO** Auto-generated method stub

**int** result;

result = arg0.getRegion().compareTo(arg1.getRegion());

**return** result;

}

}

**GenderComparator.java**

**import** java.util.Comparator;

**public** **class** GenderComparator **implements** Comparator<BankRecords> {

**public** **int** compare(BankRecords br1, BankRecords br2)

{

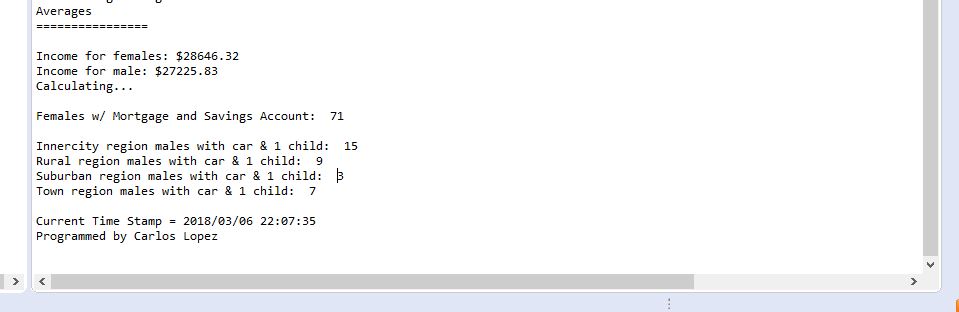
**int** result = br1.getSex().compareTo(br2.getSex());

**return** result;

}

}

**Screenshots**

****