# Bankrecords.java

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Date;

import java.util.List;

/\*\*

\* This program reads in data from bank-details.csv file and processes the data

\* then stores it in an array of objects

\* @author Chris

\*

\*/

public class BankRecords extends Client {

//variable for file name

static final String fileName = "bank-Detail.csv";

//variable for the character the csv is split by

static final String csvSplit = ",";

//Variable to count the number of lines

long numOfLines = 0;

//instantiate the array of objects

static BankRecords objects[];

//array list of a list of string

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

//instantiate all the variables

String id;

int age;

String sex;

String region;

double income;

String married;

int children;

String car;

String save\_act;

String current\_act;

String mortgage;

String pep;

/\*\*

\* @return the id

\*/

public String getId() {

return id;

}

/\*\*

\* @param id the id to set

\*/

public void setId(String id) {

this.id = id;

}

/\*\*

\* @return the age

\*/

public int getAge() {

return age;

}

/\*\*

\* @param age the age to set

\*/

public void setAge(int age) {

this.age = age;

}

/\*\*

\* @return the sex

\*/

public String getSex() {

return sex;

}

/\*\*

\* @param sex the sex to set

\*/

public void setSex(String sex) {

this.sex = sex;

}

/\*\*

\* @return the region

\*/

public String getRegion() {

return region;

}

/\*\*

\* @param region the region to set

\*/

public void setRegion(String region) {

this.region = region;

}

/\*\*

\* @return the income

\*/

public double getIncome() {

return income;

}

/\*\*

\* @param income the income to set

\*/

public void setIncome(double income) {

this.income = income;

}

/\*\*

\* @return the married

\*/

public String getMarried() {

return married;

}

/\*\*

\* @param married the married to set

\*/

public void setMarried(String married) {

this.married = married;

}

/\*\*

\* @return the children

\*/

public int getChildren() {

return children;

}

/\*\*

\* @param children the children to set

\*/

public void setChildren(int children) {

this.children = children;

}

/\*\*

\* @return the car

\*/

public String getCar() {

return car;

}

/\*\*

\* @param car the car to set

\*/

public void setCar(String car) {

this.car = car;

}

/\*\*

\* @return the save\_act

\*/

public String getSave\_act() {

return save\_act;

}

/\*\*

\* @param save\_act the save\_act to set

\*/

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

this.save\_act = save\_act;

}

/\*\*

\* @return the current\_act

\*/

public String getCurrent\_act() {

return current\_act;

}

/\*\*

\* @param current\_act the current\_act to set

\*/

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

this.current\_act = current\_act;

}

/\*\*

\* @return the mortgage

\*/

public String getMortgage() {

return mortgage;

}

/\*\*

\* @param mortgage the mortgage to set

\*/

public void setMortgage(String mortgage) {

this.mortgage = mortgage;

}

/\*\*

\* @return the pep

\*/

public String getPep() {

return pep;

}

/\*\*

\* @param pep the pep to set

\*/

public void setPep(String pep) {

this.pep = pep;

}

/\*\*

\* @return the numOfLines

\*/

public long getNumOfLines() {

return numOfLines;

}

/\*\*

\* @param numOfLines the numOfLines to set

\*/

public void setNumOfLines(long numOfLines) {

this.numOfLines = numOfLines;

}

//prints my name at the end of the console out

public void exitPrintout(){

DateFormat tf = new SimpleDateFormat("dd/MM/yy HH:mm:ss");

Date date = new Date();

System.out.println("Current Date " + tf.format(date));

System.out.println("Created By Chris Doherty \n");

}

/\*\*

\* This method will read in all the record data from the csv file and put

\* into an ArrayList

\*/

public void readData(){

String line = null;

int numLines = 0;

try {

//Wraps FileReader in BufferedReader so we can deal with one line at a time.

BufferedReader fileInput = new BufferedReader(new FileReader(fileName));

//reads each line to get the number of lines present in the file

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

//adds line into array list

array.add(Arrays.asList(line.split(csvSplit)));

numLines++;

}

//sets the number of lines in the file

setNumOfLines(numLines);

// Close file

fileInput.close();

}

catch(FileNotFoundException ex) {

System.out.println(

"Unable to open file '" +

fileName + "'");

}

catch(IOException ex) {

System.out.println(

"Error reading file '"

+ fileName + "'");

}

//calls process data method

processData();

}

/\*\*

\* This method all the record data from your ArrayList and add the data into each

\* of your instance fields via your setters

\*

\*/

public void processData() {

//casts number of lines to int and makes the array that length

int lines = (int) getNumOfLines();

objects = new BankRecords[lines];

int x = 0;

//loop to iterate through the arraylist and add to objects array

for(List<String> row: array) {

objects[x] = new BankRecords();

objects[x].setId(row.get(0));

objects[x].setAge(Integer.parseInt(row.get(1)));

objects[x].setSex(row.get(2));

objects[x].setRegion(row.get(3));

objects[x].setIncome(Double.parseDouble(row.get(4)));

objects[x].setMarried(row.get(5));

objects[x].setChildren(Integer.parseInt(row.get(6)));

objects[x].setCar(row.get(7));

objects[x].setSave\_act(row.get(8));

objects[x].setCurrent\_act(row.get(9));

objects[x].setMortgage(row.get(10));

objects[x].setPep(row.get(11));

x++;

}

//calls print data

printData();

}

/\*\*

\* This method should print the first 25 records for various fields

\* to the console via your getters

\* ID, AGE, SEX, REGION, INCOME, and MORTGAGE

\*/

public void printData() {

//makes the header for columns

System.out.println("ID\t\tAge\tSex\tRegion\t\tIncome\t\tMortgage");

//iterate through the object array and prints out the first 25

for(int i=0; i<25; i++) {

System.out.printf("%s\t\t" //ID

+ "%d\t" //Age

+ "%s\t" //Sex

+ "%-15s\t" //Region

+ "%-9s\t" //Income

+ "%s\n", //Mortgage

objects[i].getId(),objects[i].getAge(),objects[i].getSex(),

objects[i].getRegion(),objects[i].getIncome(),objects[i].getMortgage());

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

BankRecords obj = new BankRecords();

//uses an object to call read data

obj.readData();

obj.exitPrintout();

}

}

# Client.java

/\*\*

\* This program is for Lab 2 of ITMD 511

\* This abstract class allows for three abstract methods the bank needs to process.

\*

\*/

/\*\*

\* @author Chris

\*

\*/

public abstract class Client {

public void readData() {

}

public void processData(){

}

public void printData(){

}

}

# Screen Shots

Console output from reading objects array

