Practice Final Code

CSC110 Fall 2019

```
import java.io.IOException;

public class Business{
  public static void main(String [] args) throws IOException{

      ProductList pList = new ProductList("products.csv");
      TransactionList tList = new TransactionList("transactions.csv");
      RegisterList rList = new RegisterList("registers.csv", pList);

      System.out.println(pList);
      System.out.println(tList);
      System.out.println(rList);

      rList.getRegister(0).scanProduct(0);
      rList.getRegister(0).scanProduct(1);
      rList.getRegister(0).scanProduct(0);

      rList.getRegister(1).scanProduct(2);
      rList.getRegister(1).scanProduct(2);

      rList.getRegister(2).scanProduct(2);
      rList.getRegister(2).scanProduct(0);

      rList.getRegister(0).acceptPayment(100);
      rList.getRegister(1).acceptPayment(20);
      rList.getRegister(2).acceptPayment(35.75);

      for(int i = 0; i < rList.length(); i++){
        tList.insertTransaction(rList.getRegister(i).completeCheckout());
          System.out.println("Your transaction at register " + i + " is complete!");
          System.out.println("Your change is: " + rList.getRegister(i).getChange());
          System.out.println(rList.getRegister(i).getReceipt());

          System.out.println("Starting new checkout session...");
          rList.getRegister(i).startNewCheckout();
      }

      tList.saveTransactionList("transaction.csv");
      rList.saveRegisterList("registers.csv");
  }
}
```

```
public class Product{
  public Product(String name, double price){
    this.name = name;
    this.price = price;
  }

  public String getName(){
    return name;
  }

  public double getPrice(){
    return price;
  }

  public String toString(){
    return "\"" + name + "\", '\"" + price + "\"\n";
  }

  private String name;
  private double price;
  private int productId;
}
```

```
import java.io.File;
import java.util.Scanner;
import java.io.IOException;
import java.io.PrintWriter;

public class ProductList{
  public ProductList(String filename) throws IOException {
    Scanner scan = new Scanner(new File(filename));
    products = new Product[1000];
    this.length = 0;

    int i = 0;

    while(scan.hasNext()){
      String line = scan.nextLine();

      if(i != 0){
        String [] fields = line.split(",");

        String nameField = fields[0].replace("\"", "").trim();
        double priceField = Double.parseDouble(fields[1].replace("\"", "").trim());

        products[i - 1] = new Product(nameField, priceField);
      }

      i++;
    }

    if(i > 0){
      this.length = i - 1;
    }

  }

  public ProductList(){
   products = new Product[MAX_PRODUCTS];
   this.length = 0;
  }

  public ProductList(Product [] list){
    products = list;
    this.length = list.length;
  }

  public Product getProduct(int index){
    return products[index];
  }

  public double getPriceById(int id){
    return products[id].getPrice();
  }

  public String getNameById(int id){
    return products[id].getName();
  }

  public int length(){
    return length;
  }

  public void insertProduct(Product p){
    products[length] = p;
    length++;
  }

  public String toString(){
    String result = PRODUCTS_HEADER;

    for(int i = 0; i < length; i++){
      result += products[i];
    }

    return result;
  }

  public void saveProductList(String filename) throws IOException{
    File f = new File(filename);
    PrintWriter x = new PrintWriter(f);
    x.print(this);
    x.close();
  }

  public static final String CSV_START_LINE = "\"";
  public static final String CSV_SEPARATOR = "\", \"";
  public static final String CSV_END_LINE = "\"\n";
  public static final String PRODUCTS_HEADER = CSV_START_LINE +
                                               "Name" + CSV_SEPARATOR +
                                                      "Price" + CSV_END_LINE;
  public static final int MAX_PRODUCTS = 1000;
  private Product [] products;
  private int length;
}
```

```
public class Register{
  public Register(int registerId, ProductList products, double salesTax){
    this.registerId = registerId;
    this.products = products;
    this.salesTaxRate = salesTax;
    this.scannedProductIds = new int[MAX_PRODUCT_LIST_SIZE];
    this.indexOfNextProduct = 0;
    this.receipt = "\n\nReceipt\n\n";
    this.moneyInRegister = 0;
  }

  public Register(int registerId, ProductList products, double salesTax, double moneyInRegister){
    this(registerId, products, salesTax);
    this.moneyInRegister = moneyInRegister;
  }


  public double calculateTotal(){
    return salesTaxDue + subTotalDue;
  }

  public void scanProduct(int id){
    scannedProductIds[indexOfNextProduct] = id;
    indexOfNextProduct++;

    double price = products.getPriceById(id);
    Product product = products.getProduct(id);

    subTotalDue += price;
    salesTaxDue += price * salesTaxRate;

    receipt += product;

  }

  public void acceptPayment(double amount){
    this.paymentAmount = amount;
  }

  public Transaction completeCheckout(){
    Transaction t = new Transaction(registerId, paymentAmount, subTotalDue, salesTaxDue, getChange());
    moneyInRegister += (paymentAmount - getChange());

    receipt += ("\nSubTotal: " + subTotalDue + "\n");
    receipt += ("Tax at " + salesTaxRate + "% :" + salesTaxDue + "\n");
    receipt += ("Total: " + calculateTotal() + "\n\n");
    receipt += "Thank You! Come again!\n\n";

    return t;
  }


  public void startNewCheckout(){
    clearCurrentCheckout();
  }


  private void clearCurrentCheckout(){
    scannedProductIds = new int[MAX_PRODUCT_LIST_SIZE];
    indexOfNextProduct = 0;
    salesTaxDue = 0;
    subTotalDue = 0;
    paymentAmount = 0;
    receipt = "\n\nReceipt\n\n";
  }

  public String getReceipt(){
    return receipt;
  }

  public double getChange(){
    return paymentAmount - calculateTotal();
  }

  public String toString(){
    return Util.CSV_START_LINE + registerId + Util.CSV_SEPARATOR + salesTaxRate + Util.CSV_SEPARATOR + moneyInRegister +
Util.CSV_END_LINE;
  }

  public static final int MAX_PRODUCT_LIST_SIZE = 1000;

  private ProductList products;
  private int [] scannedProductIds;
  private int registerId;
  private double moneyInRegister;
  private double salesTaxRate;
  private int indexOfNextProduct;

  private double salesTaxDue;
  private double subTotalDue;
  private double paymentAmount;
  private String receipt;
}
```

```
import java.io.File;
import java.util.Scanner;
import java.io.IOException;
import java.io.PrintWriter;

public class RegisterList{

  public RegisterList(String filename, ProductList pList) throws IOException{
    Scanner scan = new Scanner(new File(filename));
    registers = new Register[MAX_REGISTERS];
    this.length = 0;

    int i = 0;

    while(scan.hasNext()){
      String line = scan.nextLine();

      if(i !=0){
        String [] fields  = line.split(",");

          int registerId = Util.cleanFieldToInt(fields[0]);
        double salesTaxRate = Util.cleanFieldToDouble(fields[1]);
          double moneyInRegister = Util.cleanFieldToDouble(fields[2]);

          registers[i - 1] = new Register(registerId, pList, salesTaxRate, moneyInRegister);
      }

      i++;
    }

    if(i > 0){
     this.length = i - 1;
    }
  }

  public void insertRegister(Register r){
    registers[length] = r;
    length++;
  }

  public Register getRegister(int index){
    return registers[index];
  }


  public String toString(){
    String result = REGISTERS_HEADER;

    for(int i = 0; i < length; i++){
      result += registers[i];
    }

    return result;
  }

  public int length(){
    return length;
  }

  public void saveRegisterList(String filename) throws IOException{
    File f = new File(filename);
    PrintWriter x = new PrintWriter(f);
    x.print(this);
    x.close();
  }

  private Register [] registers;
  private int length;

  public static final int MAX_REGISTERS = 100;
  public static final String REGISTERS_HEADER = Util.CSV_START_LINE + "Register Id" + Util.CSV_SEPARATOR + "Sales Tax Rate" +
Util.CSV_SEPARATOR + "Money in Register" + Util.CSV_END_LINE;
}
```

```
import java.sql.Timestamp;
import java.time.Instant;

public class Transaction{

  public Transaction(int registerId, double amountPaid, double subTotal, double taxPaid, double changeGiven){
    this.registerId = registerId;
    this.momentOfPurchase = new Timestamp(System.currentTimeMillis());
    this.amountPaid = amountPaid;
    this.subTotal = subTotal;
    this.taxPaid = taxPaid;
    this.changeGiven = changeGiven;
  }

  public Transaction(Timestamp t, int registerId, double amountPaid, double subTotal, double taxPaid, double changeGiven){
    this(registerId, amountPaid, subTotal, taxPaid, changeGiven);
    momentOfPurchase = t;
  }

  public Transaction(String transactionStr){
    //break apart the string and construct the object
  }

  public int getRegisterId(){
    return registerId;
  }

  public String getTimestamp(){
    return momentOfPurchase.toString();
  }

  public double getAmountPaid(){
    return amountPaid;
  }

  public double getSubTotal(){
    return subTotal;
  }

  public double getTaxPaid(){
    return taxPaid;
  }


  public String toString(){
    return CSV_START_LINE + getTimestamp() + CSV_SEPARATOR +
                              registerId + CSV_SEPARATOR +
                                  subTotal + CSV_SEPARATOR +
                                  taxPaid + CSV_SEPARATOR +
                                  amountPaid + CSV_SEPARATOR +
                                  changeGiven + CSV_END_LINE;
  }

  public static final String CSV_START_LINE = "\"";
  public static final String CSV_SEPARATOR = "\", \"";
  public static final String CSV_END_LINE = "\"\n";
  public static final String TRANSACTION_CSV_HEADING = CSV_START_LINE +
                                                       "Moment of Purchase" + CSV_SEPARATOR +
                                                               "Register ID" + CSV_SEPARATOR +
                                                               "SubTotal" + CSV_SEPARATOR +
                                                               "Tax Paid" + CSV_SEPARATOR +
                                                               "Amount Paid" + CSV_SEPARATOR +
                                                               "Change Given" + CSV_END_LINE;


  private Timestamp momentOfPurchase;
  private int registerId;
  private double amountPaid;
  private double changeGiven;
  private double subTotal;
  private double taxPaid;
}
```

```
import java.io.File;
import java.sql.Timestamp;
import java.util.Scanner;
import java.io.IOException;
import java.io.PrintWriter;

public class TransactionList{
  public TransactionList(String filename) throws IOException {
    Scanner scan = new Scanner(new File(filename));
    transactions = new Transaction[MAX_TRANSACTIONS];
    this.length = 0;

    int i = 0;

    while(scan.hasNext()){
      String line = scan.nextLine();

      if(i != 0){
        String [] fields = line.split(",");

          Timestamp  momentOfPurchase = Util.convertStringToTimestamp(fields[0].replace("\"", ""));
          int registerId = Util.cleanFieldToInt(fields[1]);
          double amountPaid = Util.cleanFieldToDouble(fields[2]);
          double changeGiven = Util.cleanFieldToDouble(fields[3]);
          double subTotal = Util.cleanFieldToDouble(fields[4]);
          double taxPaid = Util.cleanFieldToDouble(fields[5]);

          transactions[i - 1] = new Transaction(momentOfPurchase, registerId, amountPaid, changeGiven, subTotal, taxPaid);

      }

      i++;
    }

    if(i > 0){
      this.length = i - 1;
    }

  }

  public Transaction getTransaction(int index){
    return transactions[index];
  }

  public void insertTransaction(Transaction t){
    transactions[length] = t;
    length++;
  }

  public int length(){
    return length;
  }

  public String toString(){
    String result = Transaction.TRANSACTION_CSV_HEADING;

    for(int i = 0; i < length; i++){
      result += transactions[i];
    }
    return result;

  }

  public void saveTransactionList(String filename) throws IOException{
    File f = new File(filename);
    PrintWriter x = new PrintWriter(f);
    x.print(this);
    x.close();
  }

  public static final int MAX_TRANSACTIONS = 11600;
  private Transaction [] transactions;
  private int length;

}
```

```
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Util {
  public static Timestamp convertStringToTimestamp(String strDate) {
    try {
      DateFormat formatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");
      Date date = formatter.parse(strDate);
      Timestamp timeStampDate = new Timestamp(date.getTime());

      return timeStampDate;

    } catch (ParseException e) {
      System.out.println("Exception :" + e);

      return null;
    }
  }

  public static double cleanFieldToDouble(String field){
    return Double.parseDouble(field.replace("\"", "").trim());
  }

  public static int cleanFieldToInt(String field){
    return Integer.parseInt(field.replace("\"", "").trim());
  }

  public static String cleanField(String field){
    return field.replace("\"", "").trim();
  }

  public static final String CSV_START_LINE = "\"";
  public static final String CSV_SEPARATOR = "\", \"";
  public static final String CSV_END_LINE = "\"\n";

}
```