

Practice Final Code

CSC110 Fall 2019

```
1
2
3 import java.io.IOException;
4 public class Business{
5     public static void main(String [] args) throws IOException{
6
7         ProductList pList = new ProductList("products.csv");
8         TransactionList tList = new TransactionList("transactions.csv");
9         RegisterList rList = new RegisterList("registers.csv", pList);
10
11         System.out.println(pList);
12         System.out.println(tList);
13         System.out.println(rList);
14
15         rList.getRegister(0).scanProduct(0);
16         rList.getRegister(0).scanProduct(1);
17         rList.getRegister(0).scanProduct(0);
18
19         rList.getRegister(1).scanProduct(2);
20         rList.getRegister(1).scanProduct(2);
21
22         rList.getRegister(2).scanProduct(2);
23         rList.getRegister(2).scanProduct(0);
24
25         rList.getRegister(0).acceptPayment(100);
26         rList.getRegister(1).acceptPayment(20);
27         rList.getRegister(2).acceptPayment(35.75);
28
29         for(int i = 0; i < rList.length(); i++){
30             tList.insertTransaction(rList.getRegister(i).completeCheckout());
31             System.out.println("Your transaction at register " + i + " is complete!");
32             System.out.println("Your change is: " + rList.getRegister(i).getChange());
33             System.out.println(rList.getRegister(i).getReceipt());
34
35             System.out.println("Starting new checkout session...");
36             rList.getRegister(i).startNewCheckout();
37         }
38
39         tList.saveTransactionList("transaction.csv");
40         rList.saveRegisterList("registers.csv");
41     }
42 }
```

```
1 public class Product{
2     public Product(String name, double price){
3         this.name = name;
4         this.price = price;
5     }
6
7     public String getName(){
8         return name;
9     }
10
11     public double getPrice(){
12         return price;
13     }
14
15     public String toString(){
16         return "\"" + name + "\", \"" + price + "\"\n";
17     }
18
19     private String name;
20     private double price;
21     private int productId;
22 }
```

```

1  import java.io.File;
2  import java.util.Scanner;
3  import java.io.IOException;
4  import java.io.PrintWriter;

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

10         int i = 0;

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

13             if(i != 0){
14                 String [] fields = line.split(",");
15                 String nameField = fields[0].replace("\"", "").trim();
16                 double priceField = Double.parseDouble(fields[1].replace("\"", "").trim());

17                 products[i - 1] = new Product(nameField, priceField);
18             }

19             i++;
20         }

21         if(i > 0){
22             this.length = i - 1;
23         }

24     }

25     public ProductList(){
26         products = new Product[MAX_PRODUCTS];
27         this.length = 0;
28     }

29     public ProductList(Product [] list){
30         products = list;
31         this.length = list.length;
32     }

33     public Product getProduct(int index){
34         return products[index];
35     }

36     public double getPriceById(int id){
37         return products[id].getPrice();
38     }

39     public String getNameById(int id){
40         return products[id].getName();
41     }

42     public int length(){
43         return length;
44     }

45     public void insertProduct(Product p){
46         products[length] = p;
47         length++;
48     }

49     public String toString(){
50         String result = PRODUCTS_HEADER;
51         for(int i = 0; i < length; i++){
52             result += products[i];
53         }

54         return result;
55     }

56     public void saveProductList(String filename) throws IOException{
57         File f = new File(filename);
58         PrintWriter x = new PrintWriter(f);
59         x.print(this);
60         x.close();
61     }

62     public static final String CSV_START_LINE = "\"";
63     public static final String CSV_SEPARATOR = "\", \"";
64     public static final String CSV_END_LINE = "\"\n";
65     public static final String PRODUCTS_HEADER = CSV_START_LINE +
66         "Name" + CSV_SEPARATOR +
67         "Price" + CSV_END_LINE;

68     public static final int MAX_PRODUCTS = 1000;
69     private Product [] products;
70     private int length;
71 }

```

```

1  public class Register{
2      public Register(int registerId, ProductList products, double salesTax){
3          this.registerId = registerId;
4          this.products = products;
5          this.salesTaxRate = salesTax;
6          this.scannedProductIds = new int[MAX_PRODUCT_LIST_SIZE];
7          this.indexOfNextProduct = 0;
8          this.receipt = "\n\nReceipt\n\n";
9          this.moneyInRegister = 0;
10     }

11     public Register(int registerId, ProductList products, double salesTax, double moneyInRegister){
12         this(registerId, products, salesTax);
13         this.moneyInRegister = moneyInRegister;
14     }

15     public double calculateTotal(){
16         return salesTaxDue + subTotalDue;
17     }

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

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

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

25         receipt += product;
26     }

27     public void acceptPayment(double amount){
28         this.paymentAmount = amount;
29     }

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

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

37         return t;
38     }

39     public void startNewCheckout(){
40         clearCurrentCheckout();
41     }

42     private void clearCurrentCheckout(){
43         scannedProductIds = new int[MAX_PRODUCT_LIST_SIZE];
44         indexOfNextProduct = 0;
45         salesTaxDue = 0;
46         subTotalDue = 0;
47         paymentAmount = 0;
48         receipt = "\n\nReceipt\n\n";
49     }

50     public String getReceipt(){
51         return receipt;
52     }

53     public double getChange(){
54         return paymentAmount - calculateTotal();
55     }

56     public String toString(){
57         return Util.CSV_START_LINE + registerId + Util.CSV_SEPARATOR + salesTaxRate + Util.CSV_SEPARATOR + moneyInRegister +
58         Util.CSV_END_LINE;
59     }

60     }

61     public static final int MAX_PRODUCT_LIST_SIZE = 1000;

62     private ProductList products;
63     private int [] scannedProductIds;
64     private int registerId;
65     private double moneyInRegister;
66     private double salesTaxRate;
67     private int indexOfNextProduct;

68     private double salesTaxDue;
69     private double subTotalDue;
70     private double paymentAmount;
71     private String receipt;
72 }

```

```

1  import java.io.File;
2  import java.util.Scanner;
3  import java.io.IOException;
4  import java.io.PrintWriter;

5  public class RegisterList{

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

10         int i = 0;

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

13             if(i !=0){
14                 String [] fields = line.split(",");
15
16                 int registerId = Util.cleanFieldToInt(fields[0]);
17                 double salesTaxRate = Util.cleanFieldToDouble(fields[1]);
18                 double moneyInRegister = Util.cleanFieldToDouble(fields[2]);

19                 registers[i - 1] = new Register(registerId, pList, salesTaxRate, moneyInRegister);
20             }

21             i++;
22         }

23         if(i > 0){
24             this.length = i - 1;
25         }
26     }

27     public void insertRegister(Register r){
28         registers[length] = r;
29         length++;
30     }

31     public Register getRegister(int index){
32         return registers[index];
33     }

34     public String toString(){
35         String result = REGISTERS_HEADER;

36         for(int i = 0; i < length; i++){
37             result += registers[i];
38         }

39         return result;
40     }

41     public int length(){
42         return length;
43     }

44 }

45     public void saveRegisterList(String filename) throws IOException{
46         File f = new File(filename);
47         PrintWriter x = new PrintWriter(f);
48         x.print(this);
49         x.close();
50     }

51     private Register [] registers;
52     private int length;

53     public static final int MAX_REGISTERS = 100;
54     public static final String REGISTERS_HEADER = Util.CSV_START_LINE + "Register Id" + Util.CSV_SEPARATOR + "Sales Tax Rate" +
55     Util.CSV_SEPARATOR + "Money in Register" + Util.CSV_END_LINE;
56 }

```

```

1  import java.sql.Timestamp;
2  import java.time.Instant;

3  public class Transaction{

4      public Transaction(int registerId, double amountPaid, double subTotal, double taxPaid, double changeGiven){
5          this.registerId = registerId;
6          this.momentOfPurchase = new Timestamp(System.currentTimeMillis());
7          this.amountPaid = amountPaid;
8          this.subTotal = subTotal;
9          this.taxPaid = taxPaid;
10         this.changeGiven = changeGiven;
11     }

12     public Transaction(Timestamp t, int registerId, double amountPaid, double subTotal, double taxPaid, double changeGiven){
13         this(registerId, amountPaid, subTotal, taxPaid, changeGiven);
14         momentOfPurchase = t;
15     }

16     public Transaction(String transactionStr){
17         //break apart the string and construct the object
18     }

19     public int getRegisterId(){
20         return registerId;
21     }

22     public String getTimestamp(){
23         return momentOfPurchase.toString();
24     }

25     public double getAmountPaid(){
26         return amountPaid;
27     }

28     public double getSubTotal(){
29         return subTotal;
30     }

31     public double getTaxPaid(){
32         return taxPaid;
33     }

34     public String toString(){
35         return CSV_START_LINE + getTimestamp() + CSV_SEPARATOR +
36             registerId + CSV_SEPARATOR +
37             subTotal + CSV_SEPARATOR +
38             taxPaid + CSV_SEPARATOR +
39             amountPaid + CSV_SEPARATOR +
40             changeGiven + CSV_END_LINE;
41     }

42     public static final String CSV_START_LINE = "\"";
43     public static final String CSV_SEPARATOR = "\", \"";
44     public static final String CSV_END_LINE = "\"\n";
45     public static final String TRANSACTION_CSV_HEADING = CSV_START_LINE +
46         "Moment of Purchase" + CSV_SEPARATOR +
47         "Register ID" + CSV_SEPARATOR +
48         "SubTotal" + CSV_SEPARATOR +
49         "Tax Paid" + CSV_SEPARATOR +
50         "Amount Paid" + CSV_SEPARATOR +
51         "Change Given" + CSV_END_LINE;

52     private Timestamp momentOfPurchase;
53     private int registerId;
54     private double amountPaid;
55     private double changeGiven;
56     private double subTotal;
57     private double taxPaid;
58 }

```

```

1  import java.io.File;
2  import java.sql.Timestamp;
3  import java.util.Scanner;
4  import java.io.IOException;
5  import java.io.PrintWriter;

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

11         int i = 0;

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

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

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

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

23             }

24             i++;
25         }

26         if(i > 0){
27             this.length = i - 1;
28         }

29     }

30     public Transaction getTransaction(int index){
31         return transactions[index];
32     }

33     public void insertTransaction(Transaction t){
34         transactions[length] = t;
35         length++;
36     }

37     public int length(){
38         return length;
39     }

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

42         for(int i = 0; i < length; i++){
43             result += transactions[i];
44         }

45         return result;

46     }

47     public void saveTransactionList(String filename) throws IOException{
48         File f = new File(filename);
49         PrintWriter x = new PrintWriter(f);
50         x.print(this);
51         x.close();
52     }

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

56 }

```

```

1  import java.sql.Timestamp;
2  import java.text.DateFormat;
3  import java.text.ParseException;
4  import java.text.SimpleDateFormat;
5  import java.util.Date;

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

12             return timeStampDate;

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

15             return null;
16         }
17     }

18     public static double cleanFieldToDouble(String field){
19         return Double.parseDouble(field.replace("\\"", "").trim());
20     }

21     public static int cleanFieldToInt(String field){
22         return Integer.parseInt(field.replace("\\"", "").trim());
23     }

24     public static String cleanField(String field){
25         return field.replace("\\"", "").trim();
26     }

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

30 }

```