ACCELERATED CS 1&2 – CSC 250 (FALL 2015)

Project-3: Shopping

Jhon Mesa

John Kubach

Ralph Carlo Quinto

Instructor

Vinayak Elangovan, Ph.D.

Department of Computer Science

Table of Contents

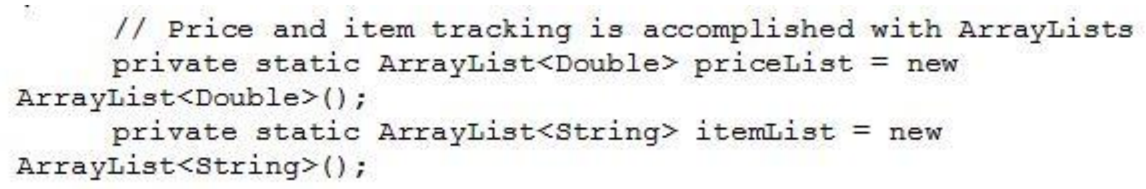
1. **Introduction**

The Java Language is a very complex and useful language in the coding world. It is a flexible language that allows for many different kinds of operations to be carried and a lot of different kinds of applications to be made. Java contains useful tools such as the Collections API, Collections algorithms, Non-linear Data Structures and Object-Oriented Principles to help with such flexibility

2. **Background**

- Java Collections API

  The Java Collections API is a set of classes that implement commonly reusable data structures, such as lists, stacks, queues, sets, and maps. Each of these different data structures holds data in a different way and each one has different methods that can be called to do different things. The following is an example of an ArrayList:



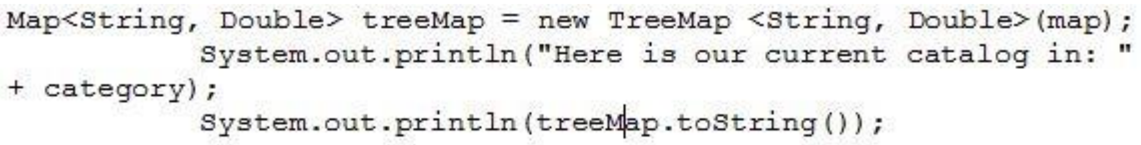
```
      // Price and item tracking is accomplished with ArrayLists
      private static ArrayList<Double> priceList = new
ArrayList<Double>();
      private static ArrayList<String> itemList = new
ArrayList<String>();
```

- Collections Algorithms

  Collections algorithms are special kinds of helper algorithms that allows to traverse a data structure and look for an element or add an element in a certain place. In our project we just converted a HashMap into a TreeMap and this is what sorted our data structure.

- Non-Linear Data Structures

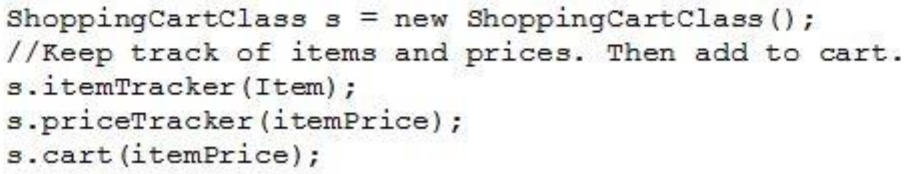
Non-Linear Data Structures are part of the Java Collections API and their data elements are not organized in a sequential fashion. Examples of Non-Linear Data Structures include multidimensional arrays, trees and graphs.



```
Map<String, Double> treeMap = new TreeMap <String, Double>(map);
           System.out.println("Here is our current catalog in: "
+ category);
           System.out.println(treeMap.toString());
```

- Object-Oriented Principles

This is one of the main components of the Java programming language. It is based on the concepts of objects, which are data structures that contain data within themselves. These objects can be manipulated through the use of the many methods that are provided in the Java API.

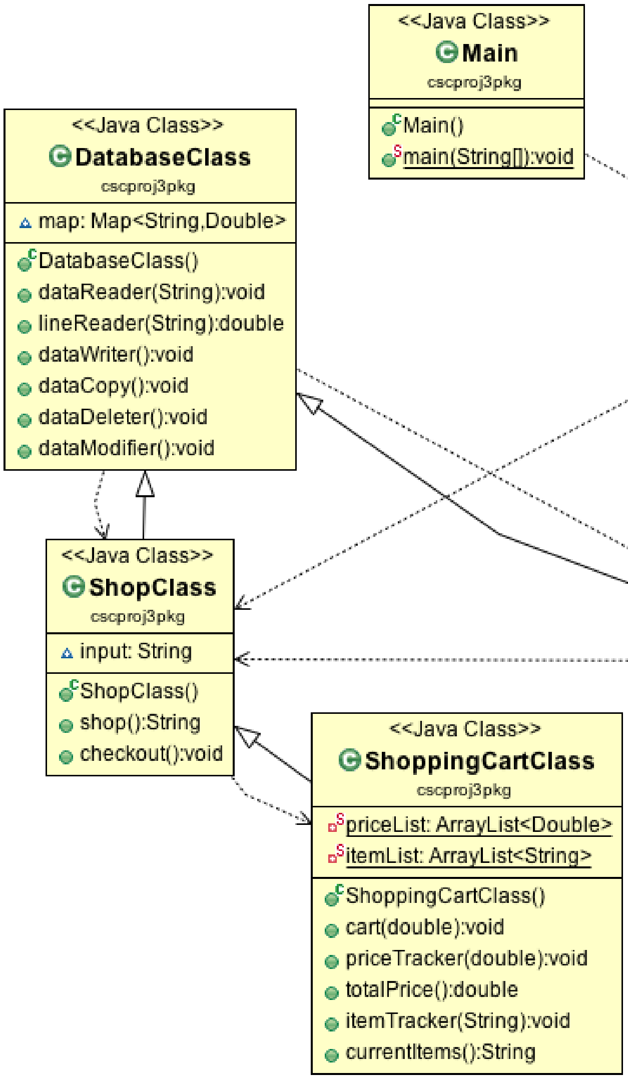


```
ShoppingCartClass s = new ShoppingCartClass();
//Keep track of items and prices. Then add to cart.
s.itemTracker(Item);
s.priceTracker(itemPrice);
s.cart(itemPrice);
```

In the example above ShoppingCartClass is instantiated and a new object s is created, which is then manipulated through the use of methods from that class.

3. Design and Implementation

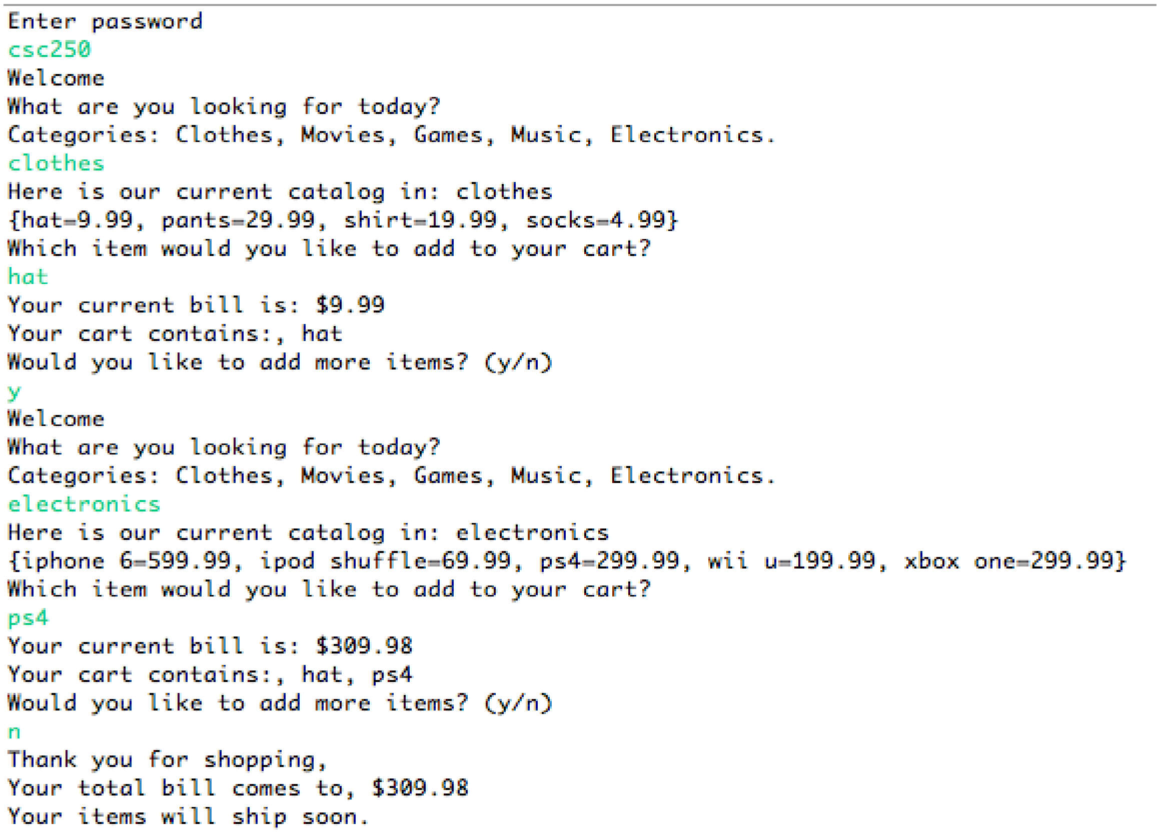
The following is a UML diagram of our program. Our program implements all sorts Object-Oriented Programming principles like inheritance, and class implementations.



In our program we also used a HashMap to store our products, and then converted that into a TreeMap to sort our items in alphabetical order.

4. Sample Outputs

Outputs for the project depend on which items are chosen to be bought. Here are a few example outputs



```
Enter password
csc250
Welcome
What are you looking for today?
Categories: Clothes, Movies, Games, Music, Electronics.
clothes
Here is our current catalog in: clothes
{hat=9.99, pants=29.99, shirt=19.99, socks=4.99}
Which item would you like to add to your cart?
hat
Your current bill is: $9.99
Your cart contains:, hat
Would you like to add more items? (y/n)
y
Welcome
What are you looking for today?
Categories: Clothes, Movies, Games, Music, Electronics.
electronics
Here is our current catalog in: electronics
{iphone 6=599.99, ipod shuffle=69.99, ps4=299.99, wii u=199.99, xbox one=299.99}
Which item would you like to add to your cart?
ps4
Your current bill is: $309.98
Your cart contains:, hat, ps4
Would you like to add more items? (y/n)
n
Thank you for shopping,
Your total bill comes to, $309.98
Your items will ship soon.
```

5. Time Complexity Analysis

In tour code, we implemented a HashMap and then converted that into a sorted TreeMap. We didn’t implement any sorting algorithms in our project.

6. Conclusion

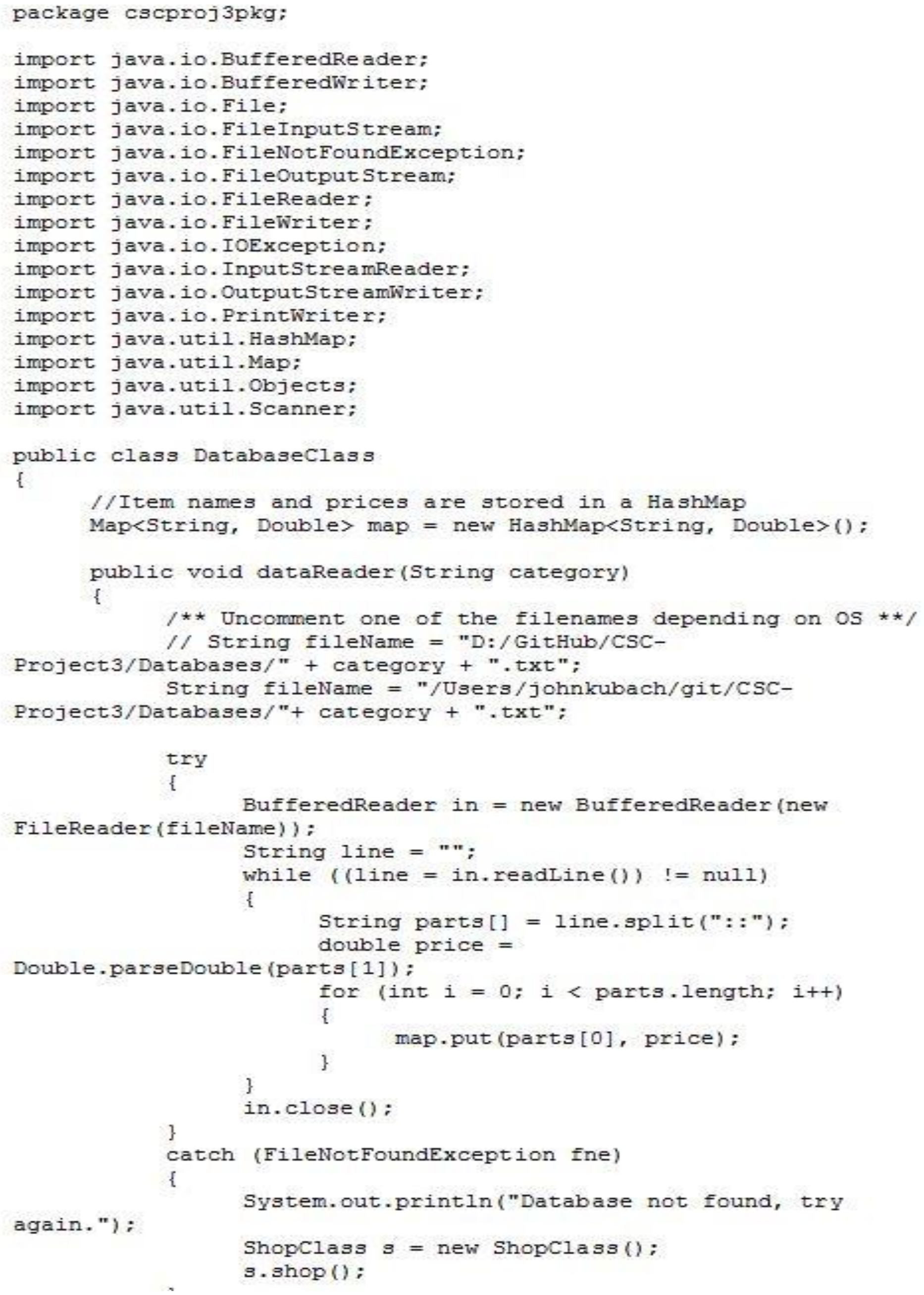
Java is a language that has many useful tools to help code a great program. It contains things like collections, methods and objects that can take coding to a different level and allow us to do all different kinds of things.

7. References

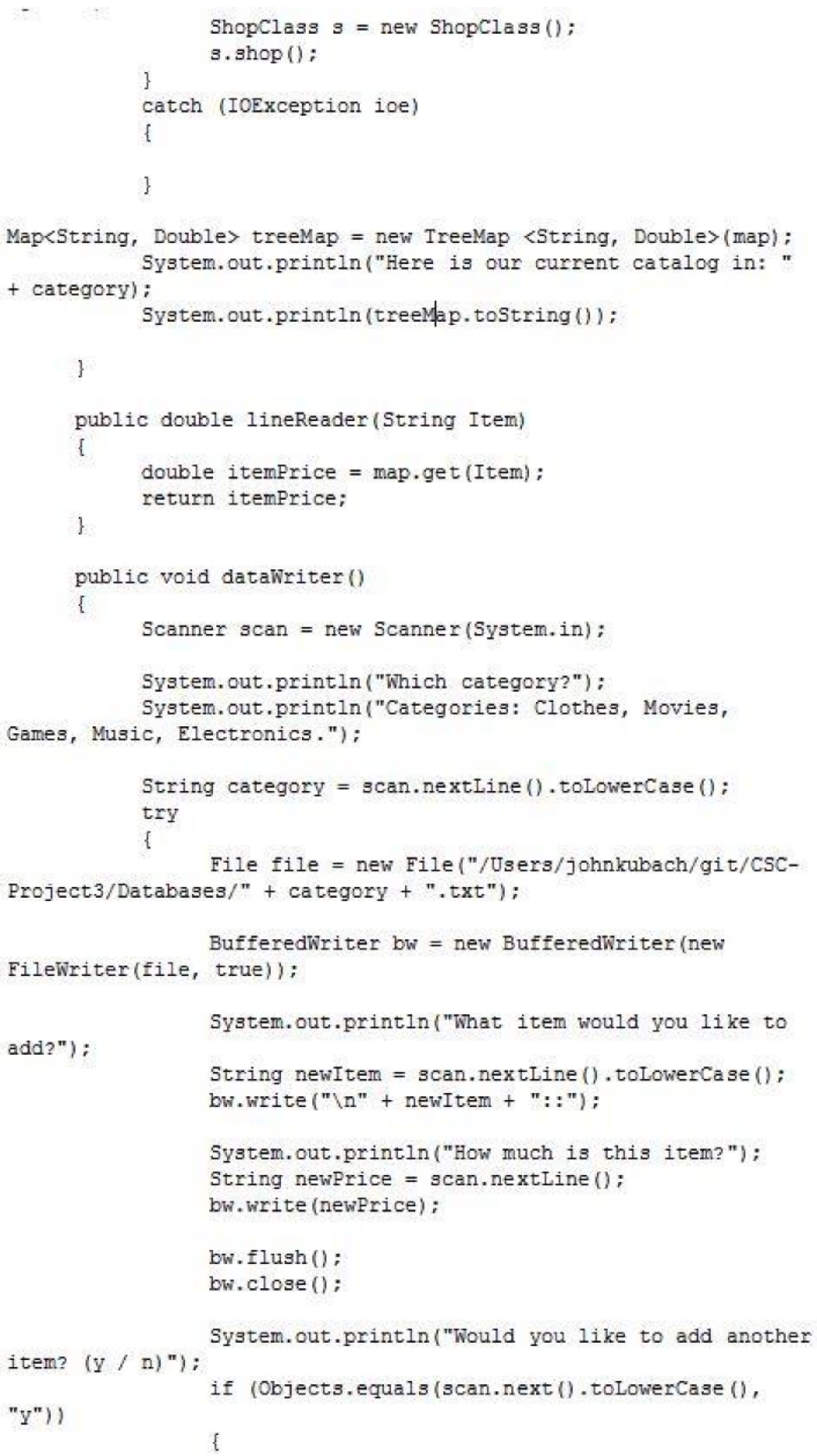
No outside source other than in-class notes and lectures.

8. Appendix

Program Codes

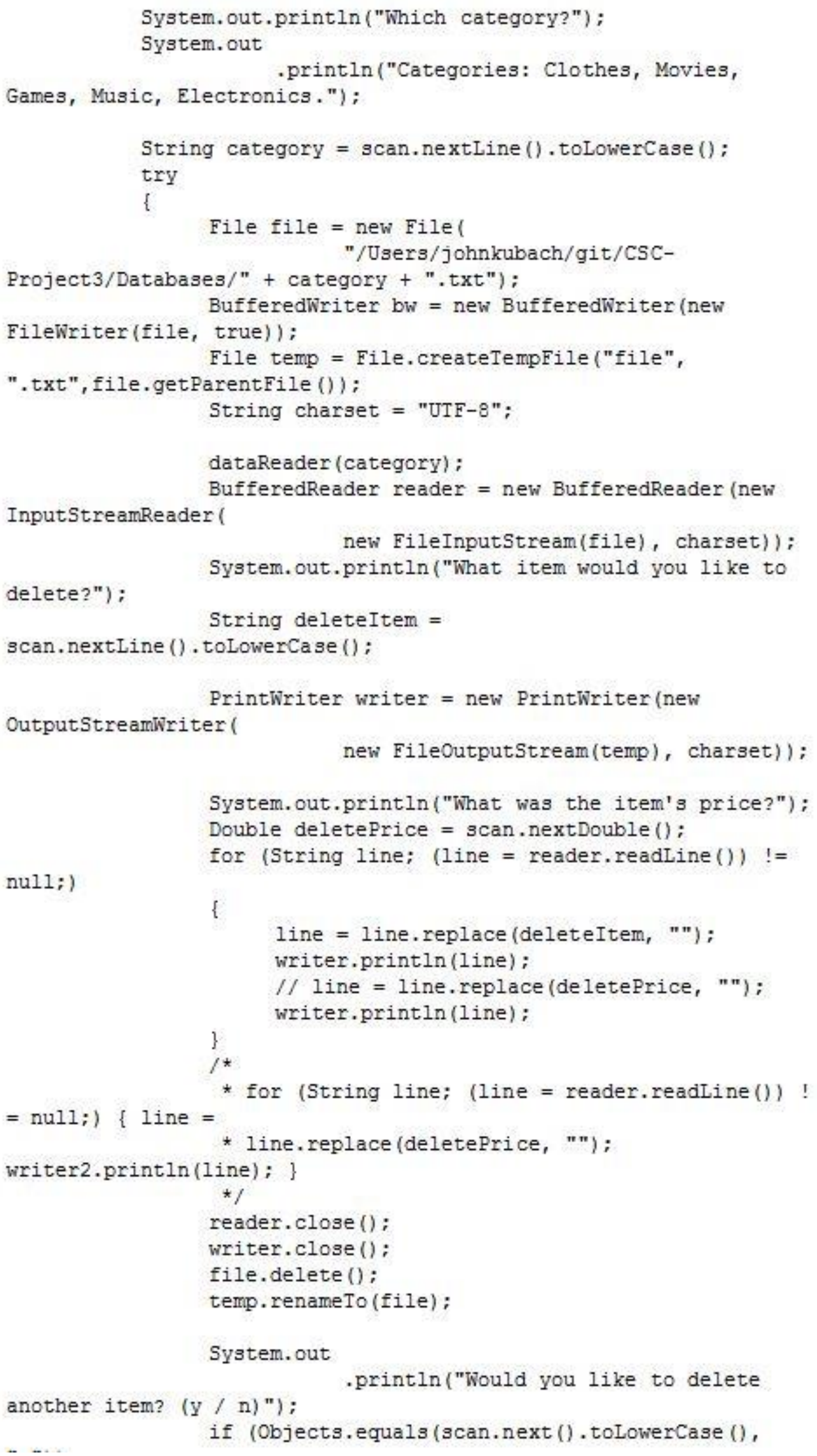


```
package cscproj3pkg;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.util.HashMap;
import java.util.Map;
import java.util.Objects;
import java.util.Scanner;

public class DatabaseClass
{
      //Item names and prices are stored in a HashMap
      Map<String, Double> map = new HashMap<String, Double>();

      public void dataReader(String category)
      {
            /** Uncomment one of the filenames depending on OS **/
            // String fileName = "D:/GitHub/CSC-
Project3/Databases/" + category + ".txt";
            String fileName = "/Users/johnkubach/git/CSC-
Project3/Databases/"+ category + ".txt";

            try
            {
                  BufferedReader in = new BufferedReader(new
FileReader(fileName));
                  String line = "";
                  while ((line = in.readLine()) != null)
                  {
                        String parts[] = line.split("::");
                        double price =
Double.parseDouble(parts[1]);
                        for (int i = 0; i < parts.length; i++)
                        {
                              map.put(parts[0], price);
                        }
                  }
                  in.close();
            }
            catch (FileNotFoundException fne)
            {
                  System.out.println("Database not found, try
again.");
                  ShopClass s = new ShopClass();
                  s.shop();
```

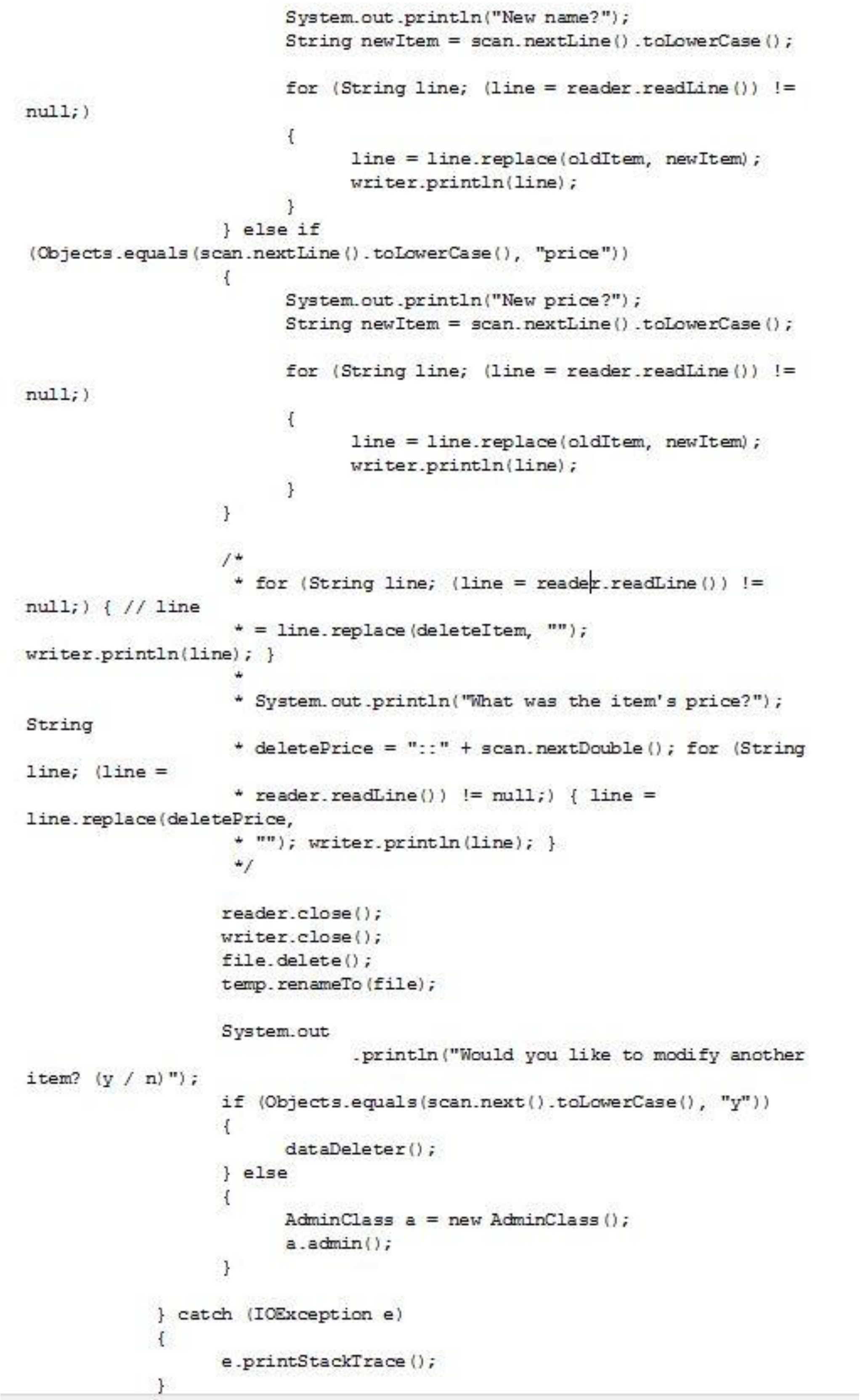


```
                ShopClass s = new ShopClass();
                s.shop();
            }
            catch (IOException ioe)
            {

            }

Map<String, Double> treeMap = new TreeMap <String, Double>(map);
            System.out.println("Here is our current catalog in: "
+ category);
            System.out.println(treeMap.toString());

        }

        public double lineReader(String Item)
        {
            double itemPrice = map.get(Item);
            return itemPrice;
        }

        public void dataWriter()
        {
            Scanner scan = new Scanner(System.in);

            System.out.println("Which category?");
            System.out.println("Categories: Clothes, Movies,
Games, Music, Electronics.");

            String category = scan.nextLine().toLowerCase();
            try
            {
                File file = new File("/Users/johnkubach/git/CSC-
Project3/Databases/" + category + ".txt");

                BufferedWriter bw = new BufferedWriter(new
FileWriter(file, true));

                System.out.println("What item would you like to
add?");
                String newItem = scan.nextLine().toLowerCase();
                bw.write("\n" + newItem + "::");

                System.out.println("How much is this item?");
                String newPrice = scan.nextLine();
                bw.write(newPrice);

                bw.flush();
                bw.close();

                System.out.println("Would you like to add another
item? (y / n)");
                if (Objects.equals(scan.next().toLowerCase(),
"y"))
                {
```

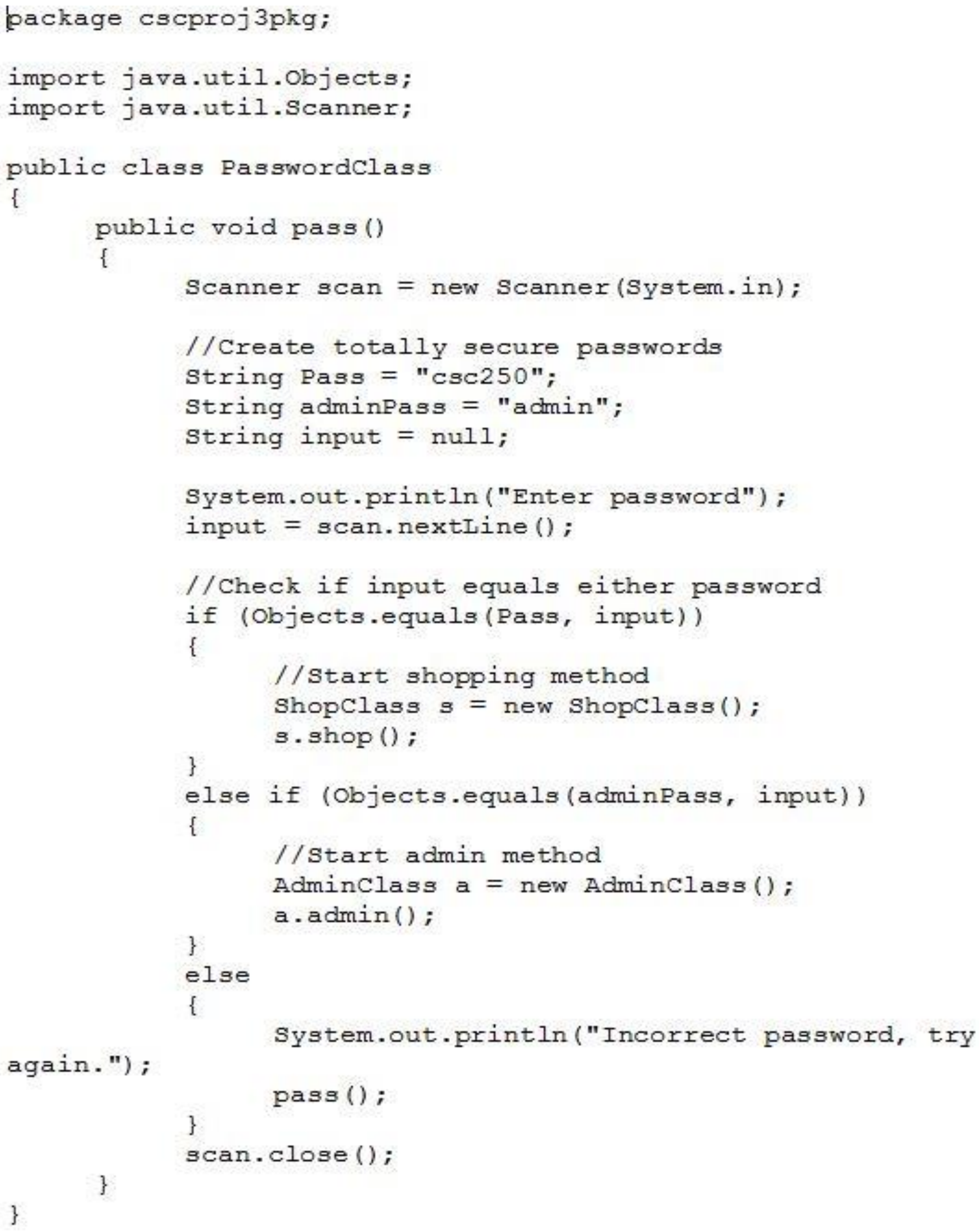
```
            System.out.println("Database not found, try
again.");
            dataWriter();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
        scan.close();
    }

    public void dataCopy()
    {
        Scanner scan = new Scanner(System.in);

        System.out.println("Which category?");
        System.out.println("Categories: Clothes, Movies,
Games, Music, Electronics.");

        String category = scan.nextLine().toLowerCase();
        try
        {

        File file = new File("/Users/johnkubach/git/CSC-
Project3/Databases/" + category + ".txt");

        File copy = File.createTempFile(category + "copy",
".txt",file.getParentFile());
        }
        catch (FileNotFoundException fne)
        {
            System.out.println("Database not found, try
again.");
            dataCopy();
        }
        catch (IOException e)
        {
        e.printStackTrace();
        }
        scan.close();
    }

    /**
     * The code below is the unfinished admin features. Deleting
and modifying
     * database entries are partially working, but both are
unable to recognize
     * the price of an item.
     **/

    public void dataDeleter()
    {
        Scanner scan = new Scanner(System.in);

        System.out.println("Which category?");
```



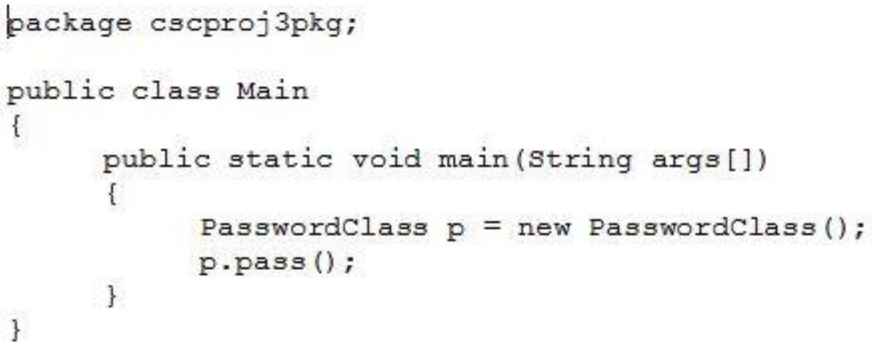
```
           System.out.println("Which category?");
           System.out
                      .println("Categories: Clothes, Movies,
Games, Music, Electronics.");

           String category = scan.nextLine().toLowerCase();
           try
           {
                File file = new File(
                           "/Users/johnkubach/git/CSC-
Project3/Databases/" + category + ".txt");
                BufferedWriter bw = new BufferedWriter(new
FileWriter(file, true));
                File temp = File.createTempFile("file",
".txt",file.getParentFile());
                String charset = "UTF-8";

                dataReader(category);
                BufferedReader reader = new BufferedReader(new
InputStreamReader(
                           new FileInputStream(file), charset));
                System.out.println("What item would you like to
delete?");
                String deleteItem =
scan.nextLine().toLowerCase();

                PrintWriter writer = new PrintWriter(new
OutputStreamWriter(
                           new FileOutputStream(temp), charset));

                System.out.println("What was the item's price?");
                Double deletePrice = scan.nextDouble();
                for (String line; (line = reader.readLine()) !=
null;)
                {
                     line = line.replace(deleteItem, "");
                     writer.println(line);
                     // line = line.replace(deletePrice, "");
                     writer.println(line);
                }
                /*
                 * for (String line; (line = reader.readLine()) !
= null;) { line =
                 * line.replace(deletePrice, "");
writer2.println(line); }
                 */
                reader.close();
                writer.close();
                file.delete();
                temp.renameTo(file);

                System.out
                           .println("Would you like to delete
another item? (y / n)");
                if (Objects.equals(scan.next().toLowerCase(),
                  if (Objects.equals(scan.next().toLowerCase(),
"y"))
                  {
                        dataDeleter();
                  } else
                  {
                        AdminClass a = new AdminClass();
                        a.admin();
                  }

            } catch (IOException e)
            {
                  e.printStackTrace();
            }
      }

      public void dataModifier()
      {
            Scanner scan = new Scanner(System.in);

            System.out.println("Which category?");
            System.out
                        .println("Categories: Clothes, Movies,
Games, Music, Electronics.");

            String category = scan.nextLine().toLowerCase();
            try
            {
                  File file = new File(
                              "/Users/johnkubach/git/CSC-
Project3/Databases/" + category
                                          + ".txt");
                  BufferedWriter bw = new BufferedWriter(new
FileWriter(file, true));
                  File temp = File.createTempFile("file", ".txt",
                              file.getParentFile());
                  String charset = "UTF-8";

                  dataReader(category);
                  BufferedReader reader = new BufferedReader(new
InputStreamReader(
                              new FileInputStream(file), charset));
                  System.out.println("What item would you like to
modify?");
                  String oldItem = scan.nextLine().toLowerCase();

                  System.out.println("Modify name, price, or
both?");

                  PrintWriter writer = new PrintWriter(new
OutputStreamWriter(
                              new FileOutputStream(temp), charset));

                  if (Objects.equals(scan.nextLine().toLowerCase(),
"name"))
```



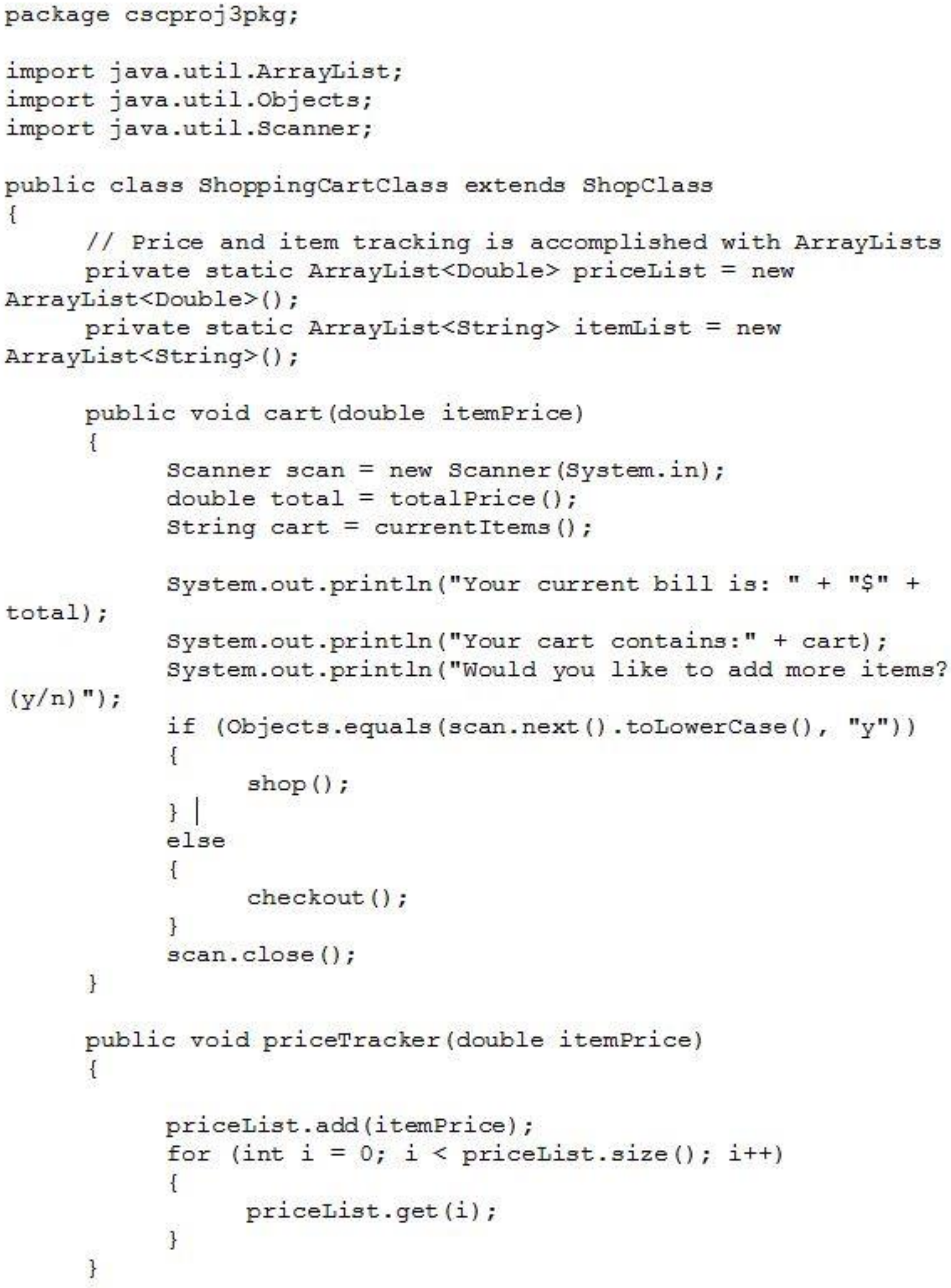
```
                        System.out.println("New name?");
                        String newItem = scan.nextLine().toLowerCase();

                        for (String line; (line = reader.readLine()) !=
null;)
                        {
                              line = line.replace(oldItem, newItem);
                              writer.println(line);
                        }
                  } else if
(Objects.equals(scan.nextLine().toLowerCase(), "price"))
                  {
                        System.out.println("New price?");
                        String newItem = scan.nextLine().toLowerCase();

                        for (String line; (line = reader.readLine()) !=
null;)
                        {
                              line = line.replace(oldItem, newItem);
                              writer.println(line);
                        }
                  }

                  /*
                   * for (String line; (line = reader.readLine()) !=
null;) { // line
                   * = line.replace(deleteItem, "");
writer.println(line); }
                   *
                   * System.out.println("What was the item's price?");
String
                   * deletePrice = "::" + scan.nextDouble(); for (String
line; (line =
                   * reader.readLine()) != null;) { line =
line.replace(deletePrice,
                   * ""); writer.println(line); }
                   */

                  reader.close();
                  writer.close();
                  file.delete();
                  temp.renameTo(file);

                  System.out
                              .println("Would you like to modify another
item? (y / n)");
                  if (Objects.equals(scan.next().toLowerCase(), "y"))
                  {
                        dataDeleter();
                  } else
                  {
                        AdminClass a = new AdminClass();
                        a.admin();
                  }

            } catch (IOException e)
            {
                  e.printStackTrace();
            }
```



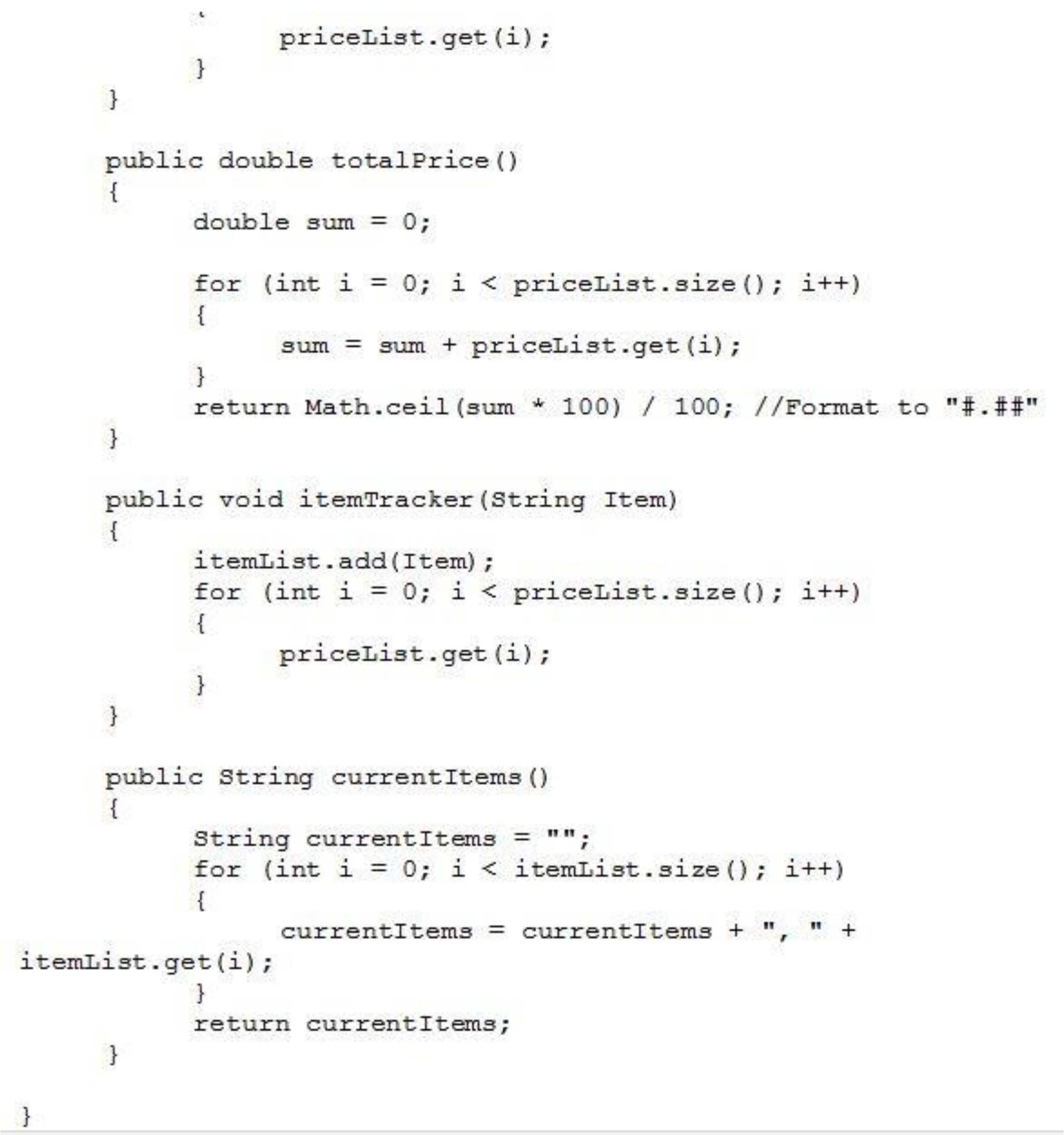
```
package cscproj3pkg;

import java.util.Objects;
import java.util.Scanner;

public class PasswordClass
{
      public void pass()
      {
            Scanner scan = new Scanner(System.in);

            //Create totally secure passwords
            String Pass = "csc250";
            String adminPass = "admin";
            String input = null;

            System.out.println("Enter password");
            input = scan.nextLine();

            //Check if input equals either password
            if (Objects.equals(Pass, input))
            {
                  //Start shopping method
                  ShopClass s = new ShopClass();
                  s.shop();
            }
            else if (Objects.equals(adminPass, input))
            {
                  //Start admin method
                  AdminClass a = new AdminClass();
                  a.admin();
            }
            else
            {
                  System.out.println("Incorrect password, try
again.");
                  pass();
            }
            scan.close();
      }
}
```



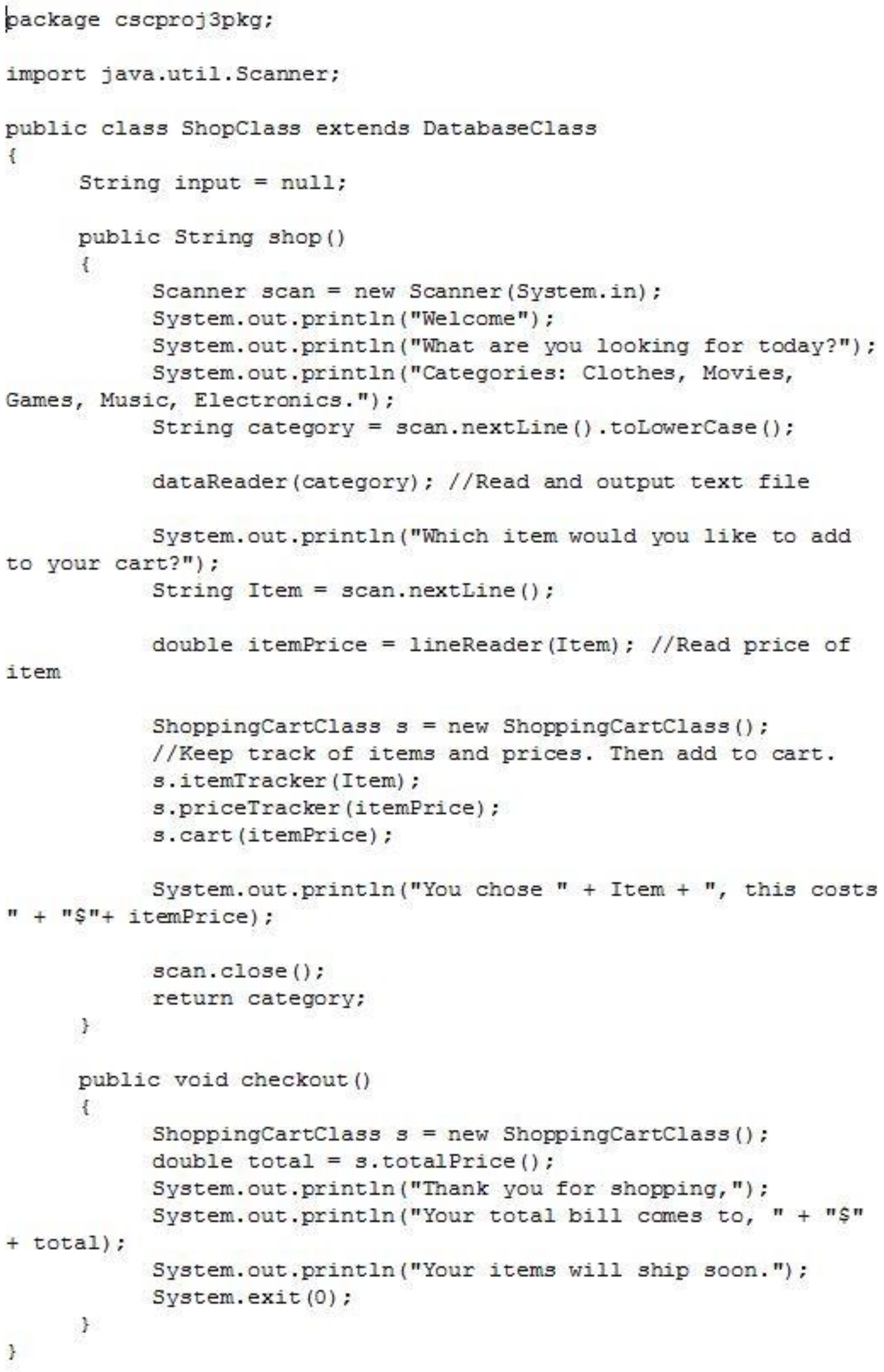
```
package cscproj3pkg;

public class Main
{
      public static void main(String args[])
      {
            PasswordClass p = new PasswordClass();
            p.pass();
      }
}
```



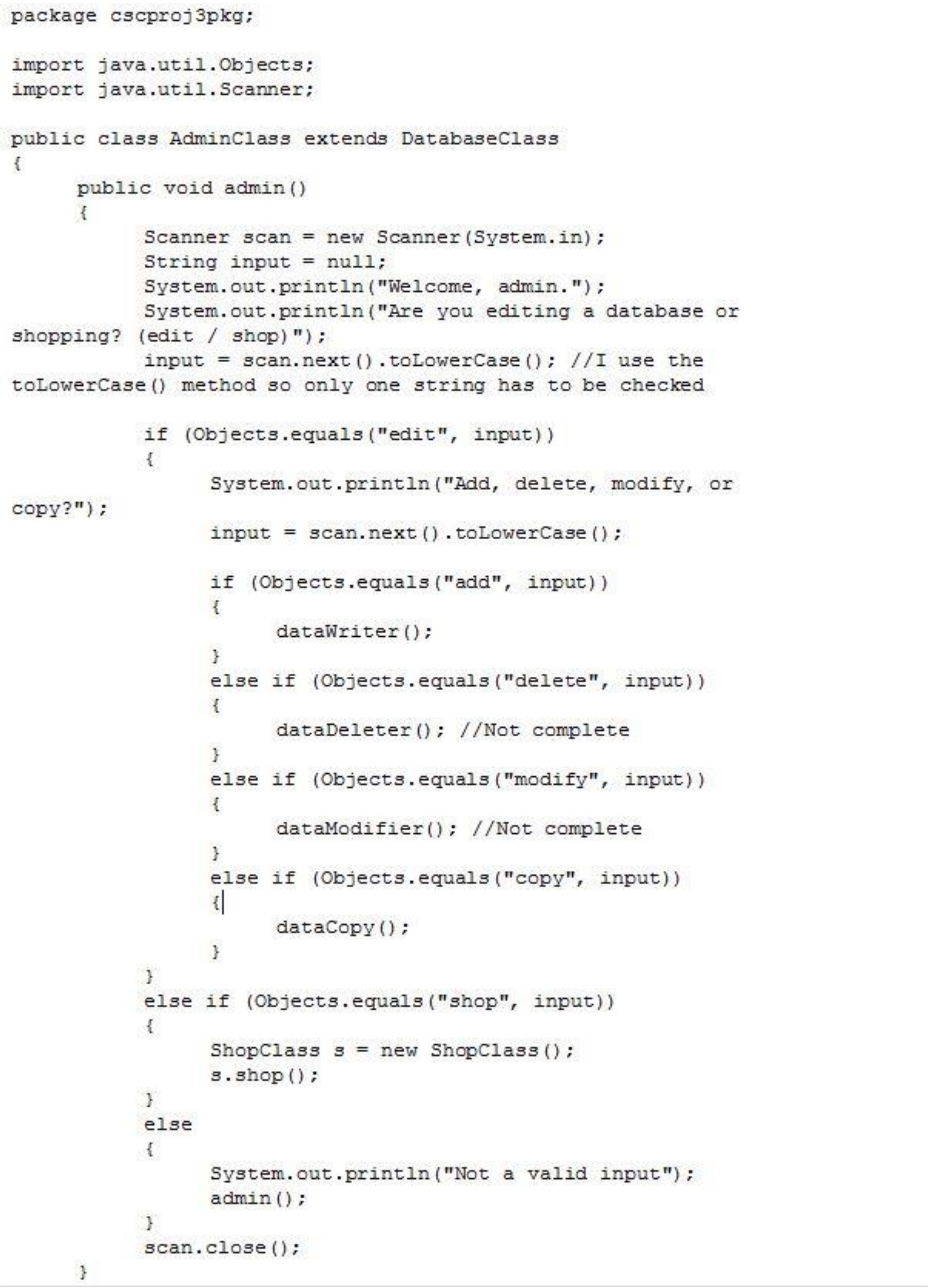
```
package cscproj3pkg;

import java.util.ArrayList;
import java.util.Objects;
import java.util.Scanner;

public class ShoppingCartClass extends ShopClass
{
      // Price and item tracking is accomplished with ArrayLists
      private static ArrayList<Double> priceList = new
ArrayList<Double>();
      private static ArrayList<String> itemList = new
ArrayList<String>();

      public void cart(double itemPrice)
      {
            Scanner scan = new Scanner(System.in);
            double total = totalPrice();
            String cart = currentItems();

            System.out.println("Your current bill is: " + "$" +
total);
            System.out.println("Your cart contains:" + cart);
            System.out.println("Would you like to add more items?
(y/n)");
            if (Objects.equals(scan.next().toLowerCase(), "y"))
            {
                  shop();
            }
            else
            {
                  checkout();
            }
            scan.close();
      }

      public void priceTracker(double itemPrice)
      {

            priceList.add(itemPrice);
            for (int i = 0; i < priceList.size(); i++)
            {
                  priceList.get(i);
            }
      }
```



```
            priceList.get(i);
        }
    }

    public double totalPrice()
    {
        double sum = 0;

        for (int i = 0; i < priceList.size(); i++)
        {
            sum = sum + priceList.get(i);
        }
        return Math.ceil(sum * 100) / 100; //Format to "#.##"
    }

    public void itemTracker(String Item)
    {
        itemList.add(Item);
        for (int i = 0; i < priceList.size(); i++)
        {
            priceList.get(i);
        }
    }

    public String currentItems()
    {
        String currentItems = "";
        for (int i = 0; i < itemList.size(); i++)
        {
            currentItems = currentItems + ", " +
itemList.get(i);
        }
        return currentItems;
    }

}
```



```
package cscproj3pkg;

import java.util.Scanner;

public class ShopClass extends DatabaseClass
{
      String input = null;

      public String shop()
      {
            Scanner scan = new Scanner(System.in);
            System.out.println("Welcome");
            System.out.println("What are you looking for today?");
            System.out.println("Categories: Clothes, Movies,
Games, Music, Electronics.");
            String category = scan.nextLine().toLowerCase();

            dataReader(category); //Read and output text file

            System.out.println("Which item would you like to add
to your cart?");
            String Item = scan.nextLine();

            double itemPrice = lineReader(Item); //Read price of
item

            ShoppingCartClass s = new ShoppingCartClass();
            //Keep track of items and prices. Then add to cart.
            s.itemTracker(Item);
            s.priceTracker(itemPrice);
            s.cart(itemPrice);

            System.out.println("You chose " + Item + ", this costs
" + "$"+ itemPrice);

            scan.close();
            return category;
      }

      public void checkout()
      {
            ShoppingCartClass s = new ShoppingCartClass();
            double total = s.totalPrice();
            System.out.println("Thank you for shopping,");
            System.out.println("Your total bill comes to, " + "$"
+ total);
            System.out.println("Your items will ship soon.");
            System.exit(0);
      }
}
```



```
package cscproj3pkg;

import java.util.Objects;
import java.util.Scanner;

public class AdminClass extends DatabaseClass
{
      public void admin()
      {
            Scanner scan = new Scanner(System.in);
            String input = null;
            System.out.println("Welcome, admin.");
            System.out.println("Are you editing a database or
shopping? (edit / shop)");
            input = scan.next().toLowerCase(); //I use the
toLowerCase() method so only one string has to be checked

            if (Objects.equals("edit", input))
            {
                  System.out.println("Add, delete, modify, or
copy?");
                  input = scan.next().toLowerCase();

                  if (Objects.equals("add", input))
                  {
                        dataWriter();
                  }
                  else if (Objects.equals("delete", input))
                  {
                        dataDeleter(); //Not complete
                  }
                  else if (Objects.equals("modify", input))
                  {
                        dataModifier(); //Not complete
                  }
                  else if (Objects.equals("copy", input))
                  {
                        dataCopy();
                  }
            }
            else if (Objects.equals("shop", input))
            {
                  ShopClass s = new ShopClass();
                  s.shop();
            }
            else
            {
                  System.out.println("Not a valid input");
                  admin();
            }
            scan.close();
      }
```