```java
/*
 * Written by JJ Shepherd
 */
import java.util.Scanner;
import java.io.*;
public class TacoManager
{
    private Taco[] tacos;
    public static final int DEF_SIZE = 10;
    public static final String DELIM = "\t";
    public static final int BODY_FIELD_AMT = 3;
    public static final int HEADER_FIELD_AMT = 2;

    public TacoManager()
    {
        init(DEF_SIZE);
    }
    public TacoManager(int size)
    {
        init(size);
    }
    public void init(int size)
    {
        if(size >= 1)
            tacos = new Taco[size];
        else
            tacos = new Taco[DEF_SIZE];
    }
    public void addTaco(Taco aTaco)
    {
        if(tacos[tacos.length-1] != null)
            return;
        for(int i=0;i<tacos.length;i++)
        {
            if(tacos[i] == null)
            {
                tacos[i] = aTaco;
                break;
            }
        }
        this.sortTacos();
    }
    public void removeTaco(String aName)
    {
        int removeIndex = -1;
        for(int i=0;i<tacos.length;i++)
        {
            if(tacos[i] != null && tacos[i].getName().equals(aName))
            {
                removeIndex = i;
                break;
            }
        }
        if(removeIndex == -1)
            return;
        else
        {
```

```
58              for(int i=removeIndex;i<tacos.length-1;i++)
59                  tacos[i] = tacos[i+1];
60              tacos[tacos.length-1] = null;
61          }
62      }
63      private void sortTacos()
64      {
65          //Using Bubble Sort
66          boolean swapped = true;
67          while(swapped == true)
68          {
69              swapped = false;
70              for(int i=0;i<tacos.length-1;i++)
71              {
72                  if(tacos[i+1]==null)
73                  {
74                      break;
75                  }
76                  if(tacos[i].getPrice()>tacos[i+1].getPrice())//Out of order swap!
77                  {
78                      Taco temp = tacos[i];
79                      tacos[i] = tacos[i+1];
80                      tacos[i+1] = temp;
81                      swapped = true;
82                  }
83              }
84          }
85      }
86      public void writeTacoFile(String aName)
87      {
88          try
89          {
90              PrintWriter fileWriter = new PrintWriter(new FileOutputStream(aName));
91              //Header
92              fileWriter.println("Taco Amt:"+DELIM+tacos.length);
93              //Body
94              for(Taco taco : tacos)
95              {
96                  if(taco == null)
97                      break;
98                  fileWriter.println(taco.getName()+DELIM+taco.getLocation()+DELIM+taco.getPrice
    ());
99              }
100             fileWriter.close();
101         }
102         catch(Exception e)
103         {
104             e.printStackTrace();
105         }
106     }
107     public void readTacoFile(String aName)
108     {
109         try
110         {
111             Scanner fileScanner = new Scanner(new File(aName));
112             //Read the header
113             String fileLine = fileScanner.nextLine();
```

```
114             String[] splitLines = fileLine.split(DELIM);
115             if(splitLines.length == HEADER_FIELD_AMT)
116             {
117                 int size = Integer.parseInt(splitLines[1]);
118                 init(size);
119             }
120             else
121                 return;
122             //Read the body
123             while(fileScanner.hasNextLine())
124             {
125                 fileLine = fileScanner.nextLine();
126                 splitLines = fileLine.split(DELIM);
127                 if(splitLines.length == BODY_FIELD_AMT)
128                 {
129                     String name = splitLines[0];
130                     String location = splitLines[1];
131                     double price = Double.parseDouble(splitLines[2]);
132                     Taco aTaco = new Taco(name,location,price);
133                     this.addTaco(aTaco);
134                 }
135             }
136             fileScanner.close();
137         }
138         catch(Exception e)
139         {
140             e.printStackTrace();
141         }
142     }
143     public void printTacos()
144     {
145         for(Taco taco : tacos)
146         {
147             if(taco == null)
148                 break;
149             System.out.println(taco);
150         }
151     }
152 }
153
```