

Homework Turnin

Email: rgalanos@fcps.edu
Section: 6G
Course: TJHSST APCS 2016-17
Assignment: 05-02
Receipt ID: 2f111070624d79a34234b0c569d45bbb

Turnin Successful!

The following file(s) were received:

SortingWidgets.java (3124 bytes)

```
1. //This code uses raw arrays. Change it to use ArrayList<E> instead.
2.
3. import java.io.*; //the File class
4. import java.util.*; //the Scanner class
5. import java.io.*; //the File class
6. import java.util.*; //ArrayList & the Scanner class in Java 1.5
7.
8. public class SortingWidgets
9. {
10.     public static void main(String[] args) throws Exception
11.     {
12.         ArrayList<Widget> apple = input("widget.txt");
13.         sort(apple);
14.         output(apple);
15.     }
16.     public static ArrayList<Widget> input(String filename) throws Exception
17.     {
18.         Scanner infile = new Scanner( new File(filename) );
19.         ArrayList<Widget> array = new ArrayList<Widget>();
20.         for (int k = 0; k < 20; k++) // read all data in the file
21.         {
22.             int x = infile.nextInt();
23.             int y = infile.nextInt();
24.             array.add(new Widget(x, y));
25.         }
26.         infile.close();
27.         return array;
28.     }
29.     public static void sort(ArrayList<Widget> array)
30.     {
31.         int maxPos;
32.         for(int k = 0; k < array.size(); k++)
33.         {
34.             maxPos = findMax(array, array.size() - k - 1);
35.             swap(array, maxPos, array.size() - k - 1);
36.         }
37.     }
38.     public static int findMax(ArrayList<Widget> array, int upper)
39.     {
40.         int maxPos = 0;
41.         for(int j = 1; j <= upper; j++)
42.             if(array.get(j).compareTo(array.get(maxPos)) > 0)
43.                 maxPos = j;
44.         return maxPos;
45.     }
46.     public static void swap(ArrayList<Widget> array, int a, int b)
47.     {
48.         Widget temp = array.get(a);
49.         array.set(a, array.get(b));
50.         array.set(b, temp);
51.     }
52. }
```

```

51.     }
52.     public static void output(ArrayList<Widget> array)
53.     {
54.         for(Widget k: array)    //use the for-each loop
55.             System.out.println(k);
56.     }
57. }
58.
59. //////////////////////////////////////////////////
60.
61. class Widget implements Comparable<Widget>
62. {
63.     //data fields
64.     private String myName;
65.     private int myPounds, myOunces;
66.
67.     //constructors
68.     public Widget()
69.     {
70.         myPounds = myOunces = 0;
71.     }
72.     public Widget(int x)
73.     {
74.         myPounds = x;
75.         myOunces = 0;
76.     }
77.     public Widget(int x, int y)
78.     {
79.         myPounds = x;
80.         myOunces = y;
81.     }
82.     public Widget(Widget arg)
83.     {
84.         myPounds = arg.getPounds();
85.         myOunces = arg.getOunces();
86.     }
87.
88.     //accessors and modifiers
89.     public int getPounds()
90.     {
91.         return myPounds;
92.     }
93.     public int getOunces()
94.     {
95.         return myOunces;
96.     }
97.     public void setPounds(int x)
98.     {
99.         myPounds = x;
100.    }
101.    public void setOunces(int x)
102.    {
103.        myOunces = x;
104.    }
105.
106.    //other methods
107.    public int compareTo(Widget w)
108.    {
109.        // Widget w = (Widget)arg;    no need to cast
110.        if(myPounds < w.getPounds())
111.            return -1;
112.        if(myPounds > w.getPounds())
113.            return 1;
114.        if(myOunces < w.getOunces())
115.            return -1;
116.        if(myOunces > w.getOunces())
117.            return 1;
118.        return 0;
119.    }
120.    public boolean equals(Widget arg)
121.    {
122.        return compareTo(arg) == 0;
123.    }
124.    public String toString()
125.    {
126.        return myPounds + " lbs. " + myOunces + " oz.";
127.    }
128. }

```