

```

1  /*
2   * Create an array of dog objects
3   * Read in from a text file
4   */
5  package arrayofobjects;
6
7  public class ArrayOfObjectsUI {
8
9      public static void main(String[] args) {
10         DogManager dm = new DogManager();
11
12         System.out.println("Display all from text file");
13         String text = dm.displayAll();
14         System.out.println(text);
15     }
16 }
17 =====
18
19 /*
20 * Dog Manager with methods to create the object arrays
21 * The constructor method reads in the text file
22 */
23 package arrayofobjects;
24
25 import java.io.File;
26 import java.io.FileNotFoundException;
27 import java.util.Scanner;
28 import java.util.logging.Level;
29 import java.util.logging.Logger;
30
31 public class DogManager {
32     // global variables including arrays
33     // The array is of the parent class ie only one array type
34     private Dogs[] dogArray = new Dogs[20];
35     private int size = 0;
36
37     // The constructor class
38     DogManager() {
39         try {
40             Scanner scFile = new Scanner(new File("dogs.txt"));
41
42             while(scFile.hasNext()) {
43                 String line = scFile.nextLine();
44                 Scanner scLine = new Scanner(line).useDelimiter("#");
45                 String name = scLine.next();
46                 String ageSt = scLine.next();
47                 int age = Integer.parseInt(ageSt);
48                 String pedigreeSt = scLine.next();
49                 boolean pedigree = Boolean.parseBoolean(pedigreeSt);
50                 dogArray[size] = new Dogs (name, age, pedigree);
51
52                 size = size + 1;
53             } // end while
54         } catch (FileNotFoundException ex) {
55             System.out.println("Error. File not found.");
56             Logger.getLogger(DogManager.class.getName()).log(Level.SEVERE, null, ex);
57         }
58     }
59     // -----
60
61     public String displayAll() {
62         String s = "";
63         for(int i = 0; i < size; i++) {
64             s = s + dogArray[i].toString() + "\n";
65         } // end for
66         return s;
67     }
68 } // end class
69
70 =====
71

```

```

72  /*
73  * Dog class from which objects are created
74  * You can use "insert code" in NetBeans to create your getter
75  * and setter methods.
76  */
77  package arrayofobjects;
78
79  public class Dogs {
80      private String name = null;
81      private int age = 0;
82      private boolean pedigree = false;
83
84      // The constructor method
85      Dogs(String n, int a, boolean p) {
86          name = n;
87          age = a;
88          pedigree = p;
89      }
90      // -----
91
92      public String getName() {
93          return name;
94      }
95      // -----
96
97      public void setName(String name) {
98          this.name = name;
99      }
100     // -----
101
102     public int getAge() {
103         return age;
104     }
105     // -----
106
107     public void setAge(int age) {
108         this.age = age;
109     }
110     // -----
111
112     public boolean isPedigree() {
113         return pedigree;
114     }
115     // -----
116
117     public void setPedigree(boolean pedigree) {
118         this.pedigree = pedigree;
119     }
120     // -----
121
122     public String toString() {
123
124         return name + "\t" + age + "\t" + pedigree;
125     }
126
127 }
128 =====
129
130 This is what the text file looks like . . .
131
132 Butch#2#false
133 April#1#true
134 Shaka#2#true
135 Morgan#4#false
136 Ratty#9#false
137 Ricky#1#false
138 Totum#2#true
139 Seal#1#false
140
141 =====
142

```

```
143  OUTPUT
144
145  run:
146  Display all from text file
147  Butch    2    false
148  April    1    true
149  Shaka    2    true
150  Morgan   4    false
151  Ratty    9    false
152  Ricky    1    false
153  Totum    2    true
154  Seal     1    false
155
156  BUILD SUCCESSFUL (total time: 0 seconds)
157
```