```java
/**
 * Room.java
 * for Adventure Time
 * Created by: Jessenia Aguilar-Hernandez
 * Modified By: Lauren Luo & Adrianna Valle
 *
 * Creates a Room object that contains an array of 10 keys.
 * For each room that is connected to this room, the karray
 * holds a key that can unlock the door to the connecting room.
 */
import java.util.*;

public class Room implements Comparable<Room> {
  //instance variables
  private String img;
  private Key[] karray;//holds the keys in the room
  private final int DEFAULT_CAPACITY = 10;

  /**
   * Constuctor takes in imagefile name, and the active key in the room
   */
  public Room(String imgFile, LinkedList<Key> activeKeys){
    img = imgFile;

    karray = new Key[DEFAULT_CAPACITY]; //each room will always have 10 keys
    for (int i = 0; i < activeKeys.size(); i++) {
      karray[i] = activeKeys.get(i);
    }

    for (int i = activeKeys.size(); i < karray.length; i++) {
      Key temp = GameMap.getRandomKey(); //assigns random keys to the remaining key slots in karray
      while(contains(temp)) {
        temp = GameMap.getRandomKey();
      }
      karray[i] = temp.copyKey();
    }

    /*Shuffles the key placement so that all the active keys are not
     in the beginning*/
    Random ran = new Random();
    for(int j = 0; j<karray.length; j++) {
      int swapVal = ran.nextInt(karray.length-1)+1;
      Key tempEle = karray[swapVal];
      karray[swapVal] = karray[karray.length-1-j];
      karray[karray.length-1-j] = tempEle;
    }
  }

  /**
   * Returns true if karray contains given key
   * Otherwise returns false
   */
  private boolean contains(Key kIn) {
    for (int i = 0; i < karray.length; i++) {
      if (karray[i] != null && karray[i].getName().equals(kIn.getName())) {
        return true;
      }
    }
    return false;
  }

  public String getRoomName() {
    return img.substring(0,img.indexOf("."));
  }

  /**
   * Getter: @return the keys in the room
   */
  public Key[] getRoomKeys(){
    return karray;
  }

  /**
   * Checks if the input key name is in this room's karray (is it a valid key)
   * @return boolean true if valid, else false
   */
  public boolean validKey(String keyString) {
    for (Key k: karray) {
      if (k.getName().toLowerCase().equalsIgnoreCase(keyString)) {
        return true;
      }
    }
    return false;
  }

  /**
   * Takes in name of the key and returns the Key object with that name.
   * If key doesn't exist in this room's karray, returns null.
   */
  public Key getKey(String nameOfSelectedKey) {
    for (int i = 0; i < karray.length; i++) {
      if (karray[i].getName().equalsIgnoreCase(nameOfSelectedKey)) {
        return karray[i];
```

```
94          }
95        }
96        return null;
97      }
98
99      /**
100      * compareTo: @return returns an int based on the string compareTo of room names
101      */
102     public int compareTo(Room other){
103
104       //calls on integers compareTo
105       return getRoomName().compareTo(other.getRoomName());
106     }
107
108     /**
109      * Returns String representation of the object.
110      * For testing purposes
111      */
112     public String toString() {
113       return img.substring(0,img.indexOf("r"));
114
115
116 //      String s = "name: " + getImage(); //+ "\nkey(s): ";
117 //
118 //      for (int i = 0; i < karray.length; i++) {
119 //        s += "[" + karray[i].getName() + "] ";
120 //      }
121 //
122 //      return s;
123     }
124
125     //testing
126     public static void main(String[] args){
127 //      Key k1 = new Key("apple", "jals;fjdlkf", "adfhkjsdhfj");
128 //      Key k2 = new Key("toy","jdsl;jaf", "jdsklfjds");
129 //
130 //      Room r1 = new Room("somefilename", k1);
131 //
132 //      System.out.println("6 - ");
133 //      System.out.println(r1.getKey(0).getName());
134 //
135 //      Room r2 = r1;
136 //
137 //      System.out.println("Should be 0 - "+r2.compareTo(r1));
138 //
139 //      Room r3 = new Room("somefilename", k2);
140 //      r3.addKey(k1);
141 //
142 //      System.out.println("Should be 0 - "+r2.compareTo(r3));
143 //
144 //      Room r4 = new Room("somefilename", k2);
145 //      r4.addKey(k1);
146 //      r4.addKey(k1);
147 //      System.out.println("Should be less than 0 : "+r2.compareTo(r4));
148 //
149 //      Room r5 = new Room("anotherfilename",k1);
150 //      r5.addKey(k1);
151 //      System.out.println("Should be greater than 0 ? "+r2.compareTo(r5));
152     }
153 }
```