

```

1  /*
2  * Programmer:  Chad Philip Johnson
3  * Date Created:  Thursday, November 06th, 2012
4  * Date of Last Modification:  Tuesday, December 11th, 2012
5  *
6  * Description:
7  * World.class sets all of the properties of the various the rooms in the
8  * dungeon.  It depends on Room.class to instantiate rooms in the dungeon,
9  * Item.class and its subclasses Weapon.class and Armor.class to place
10 * items within rooms, and Character.class to place characters within rooms.
11 */
12
13 import java.util.*;
14 import java.io.Serializable;
15
16 /**
17 * World.class sets all of the properties of the various the rooms in the
18 * dungeon.  It depends on Room.class to instantiate rooms in the dungeon,
19 * Item.class and its subclasses Weapon.class and Armor.class to place
20 * items within rooms, and Character.class to place characters within rooms.
21 *
22 * @author Chad Philip Johnson
23 * @version 1.0
24 */
25
26 public class World implements Serializable {
27
28     static final int NUMBER_OF_ROOMS        = 6;
29     static final int NO_DOOR                = -2;
30     static final int SEALED_DOOR           = -1;
31
32     Room[] someRooms;
33
34     /**
35     * Default constructor:
36     * Instantiate all of the dungeon's rooms and the items and characters that go within them.
37     */
38
39     public World() {
40
41         this.someRooms        = new Room[NUMBER_OF_ROOMS];
42
43         // Construct layout of the dungeon
44         this.someRooms[0]     = new Room( "Lobby",
45             "This room is medium in size and the ground is cold and firm.  Torchlight\n" +
46             "illuminates the corners of the room.  To the West is a large door bolted shut:\n" +
47             "the way back to the town has been sealed for good.  To the East is a short\n" +
48             "passageway into another room.",
49             new int[] { /* North door */ NO_DOOR,
50                 /* South door */ NO_DOOR,

```

```

51         /* West door */ NO_DOOR,
52         /* East door */ 1 },
53     new Item[] { new Weapon( "Club", 4 ), new Armor( "Leather Armor", -4 ) },
54     new Character[] { /* No characters */ },
55     12 );
56     this.someRooms[1] = new Room( "Inner Chamber",
57     "This room is large, extending in each direction for at least 50 meters.\n" +
58     "Dried up bones litter the floor and the smell of death is all about you.\n" +
59     "To the West is a short passageway. To the North and South are doors\n" +
60     "leading to different rooms. To the East is a set of impressively sized\n" +
61     "double-doors that would require a great deal of effort to open.",
62     new int[] { /* North door */ 2,
63         /* South door */ 3,
64         /* West door */ 0,
65         /* East door */ SEALED_DOOR },
66     new Item[] { /* No items */ },
67     new Character[] { new Character( "Cross-eyed Giant",
68         "I guard the treasure. This is Bob's treasure. Only
69         Bob may pass.",
70         "Oooh... that tickles.",
71         "Oh hi Bob. Let me open the door for you." ) },
72     4 );
73     this.someRooms[2] = new Room( "Barracks",
74     "This room is very small in size. There are shredded cots and garments\n" +
75     "about the room. To the South is a passageway.",
76     new int[] { /* North door */ NO_DOOR,
77         /* South door */ 1,
78         /* West door */ NO_DOOR,
79         /* East door */ NO_DOOR },
80     new Item[] { new Armor( "Shield", -2 ) },
81     new Character[] { /* No characters */ },
82     2 );
83     this.someRooms[3] = new Room( "Armory",
84     "Gravel and dirt surround your feet. The racks containing weapon and armor\n" +
85     "items have been picked clean. Whatever things of value that were once here\n" +
86     "are long gone. There is a door to the West and a passageway to the North.",
87     new int[] { /* North door */ 1,
88         /* South door */ NO_DOOR,
89         /* West door */ 4,
90         /* East door */ NO_DOOR },
91     new Item[] { /* No items */ },
92     new Character[] { /* No characters */ },
93     0 );
94     this.someRooms[4] = new Room( "Armory Storage",
95     "Here is the source of all of the dirt and gravel: the ceiling has partially\n" +
96     "caved in, spilling rock and debris everywhere. There isn't much room to move\n" +
97     "around here and the only exit is to the East.",
98     new int[] { /* North door */ NO_DOOR,
99         /* South door */ NO_DOOR,
100        /* West door */ NO_DOOR,

```

```

100         /* East door */ 3 },
101     new Item[] { new Weapon( "Sword", 8 ) },
102     new Character[] { /* No characters */ },
103     7
104     );
105     this.someRooms[5] = new Room( "Treasure Room",
106     "Mountains of gold shimmer as far as the eye can see. Countless gems glisten\n" +
107     "in the torchlight. Unfortunately you are sealed in the dungeon and can't get\n" +
108     "any gold out. Your stomach grumbles.",
109     new int[] { /* North door */ NO_DOOR,
110     /* South door */ NO_DOOR,
111     /* West door */ 1,
112     /* East door */ NO_DOOR },
113     new Item[] { /* No items */ },
114     new Character[] { /* No characters */ },
115     1000000000 );
116 }
117
118 /**
119  * Returns the default start room when beginning a new game.
120  *
121  * @return Room Room value for the correct room where the player's adventure begins.
122  */
123
124 public Room getStartRoom() { return someRooms[0]; }
125
126 /**
127  * Get the room that the player wishes to go to.
128  *
129  * @param intIndex Numerical value of the door the player went through (or the direction moved in).
130  * @return Room Room value for the connected room object that exists when travelling in that direction.
131  */
132
133 public Room getRoom( int intIndex ) { return someRooms[intIndex]; }
134
135 }

```