

```
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  * Storage.class allows the saving and loading of an instance of
8  * Game.class.  Storage.class depends on Game.class for saving and
9  * loading.
10 *
11 * NOTE:  This code was adapted from FarmStorage.class by
12 * J. Boyd Trolinger.
13 */
14
15 import java.io.*;
16
17 /**
18 * Storage.class allows the saving and loading of an instance of
19 * Game.class.  Storage.class depends on Game.class for saving and
20 * loading.
21 *
22 * NOTE:  This code was adapted from FarmStorage.class by
23 * J. Boyd Trolinger.
24 *
25 * @author J. Boyd Trolinger
26 * @author Chad Philip Johnson
27 * @version 1.0
28 */
29
30 public class Storage    {
31
32     /**
33     * Loads a saved (or serialized) instance of Game.class.
34     *
35     * @param strFilename The name of the file to open.
36     * @return Game The player's saved game from a previous session.
37     */
38
39     public static Game loadGame( String strFilename )    {
40
41         Game someGame = null;
42
43         try {
44
45             FileInputStream fis = new FileInputStream( strFilename );
46             ObjectInputStream ois = new ObjectInputStream( fis );
47             someGame = (Game) ois.readObject();
48             ois.close();
49
50         } catch ( IOException ioe ) {
```

```

51         System.err.println( "Problem loading game. Creating a new game instead.\n" );
52     }
53     } catch ( ClassNotFoundException cfe ) {
54     }
55     System.err.println( "Problem loading game. Creating a new game instead.\n" );
56     }
57     }
58     }
59     if ( someGame == null ) {     someGame = new Game( new Player( "Player" ) ); }
60     return someGame;
61 }
62 }
63 }
64 }
65 }
66 /**
67  * Saves an instance of Game.class.
68  *
69  * @param someGame The instance Game.class to be saved/serialized.
70  * @param strFilename The name of the file to create.
71  * @return boolean Returns true when the save is successful.
72  */
73
74 public static boolean saveGame ( Game someGame, String strFilename )    {
75
76     try {
77
78         FileOutputStream fos = new FileOutputStream( strFilename );
79         ObjectOutputStream oos = new ObjectOutputStream( fos );
80         oos.writeObject( someGame );
81         oos.close();
82
83         return true;
84     } catch ( IOException ioe )    {
85
86         System.err.println( "Unable to save your game.\n" );
87         System.err.println( ioe + "\n" );
88
89         return false;
90     }
91     }
92     }
93     }
94     }
95     }
96     }
97     }
98     }
99     }

```