Assignment 3

CS2212 – Introduction to Software Engineering

Group 2

Kevin Brightwell

Jonathan Demelo

Graem Littleton

Justin McDonald

Ramesh Raj

Professor Laura Ried

TA Mike Tighe

16 March 2012

Table of Contents:

Section 2:

Driver and Stub Example:

```
package test;

import java.util.ArrayList;

import data.Contestant;
import data.GameData;
import data.InvalidFieldException;


/*
 * Test Case: Admin casting off a contestant after season started
 */
public class Testing{

	//Constant values for testing purposes
	private static final Integer NUM_TEST_CONTESTANTS = 6;
	private static final Integer TEST_BET_AMOUNT = 5;
	private static final String TEST_TRIBE_ONE = "Alive";
	private static final String TEST_TRIBE_TWO = "Dead";


	//DRIVER TESTS TO SEE IF CAST OFF IS WORKING PROPERLY
	public static void driver() throws InvalidFieldException{

		//Instantiate a new GameData object with 6 test contestants.
		GameData g = new GameData(NUM_TEST_CONTESTANTS);

		//Instantiate valid tribe names for the game.
		g.setTribeNames(TEST_TRIBE_ONE, TEST_TRIBE_TWO);

		//Strings to be used to compose test Contestants, so that a season may
be started.
		String[] testID = {"MJ", "BP", "JS", "AJ", "AE", "JB"};
		String[] testFirstName = {"Michael", "Brad", "Jessica", "Angelina",
"Albert", "Al"};
		String[] testLastName = {"Jackson", "Pitt", "Simpson", "Jolie",
"Einstein", "Capone"};
		String[] testTribe = {TEST_TRIBE_TWO, TEST_TRIBE_ONE, TEST_TRIBE_ONE,
TEST_TRIBE_ONE, TEST_TRIBE_TWO, TEST_TRIBE_TWO};

		//Instantiate and add the 6 test contestants to the game.
		for (int i = 0; i < 6; i++){
			Contestant c = new Contestant(testID[i], testFirstName[i],
testLastName[i], testTribe[i]);
			g.addContestant(c);
		}

		//Start the season, so that contestants may be cast off, and the weeks
can be advanced.
		g.startSeason(TEST_BET_AMOUNT);


		// TEST 1: CAST OFF THE (TEST) CONTESTANT "JESSICA SIMPSON"
		g.getAllContestants().get(2).toCastOff();	//set the next contestant
to be cast off
		g.advanceWeek();						//advance the
week, committing the cast off

		// CONFIRMATION/OUTPUT: isCastOff returns true if contestant NOT casted
off
```

```
            if (!g.getAllContestants().get(2).isCastOff())
                  System.out.println("Test 1: FAILED\n\tJessica Simpson was not
cast off.\n");
            else System.out.println("Test 1: PASSED\n\tJessica Simpson was cast off
during Week " + g.getAllContestants().get(2).getCastDate() + "\n");


            // TEST 2: CAST OFF THE (TEST) CONTESTANT "MICHAEL JACKSON"
            g.getAllContestants().get(0).toCastOff();     //set the next contestant
to be cast off
            g.advanceWeek();                                          //advance the
week, committing the cast off

            // CONFIRMATION/OUTPUT:
            if (!g.getAllContestants().get(0).isCastOff())
                  System.out.println("Test 2: FAILED\n\tMichael Jackson was not
cast off.\n");
            else System.out.println("Test 2: PASSED\n\tMichael Jackson was cast off
during Week " + g.getAllContestants().get(0).getCastDate() + "\n");
      }

      /**************END OF DRIVER TESTING----START OF STUB
TESTING*******************/
      static //Driver for the stub
      ArrayList<Contestant> list = new ArrayList<Contestant>();

      public static void stub(){
            System.out.println("STUB TESTING\n\n\n");
            //add 6(max number) of contestants
            try {
                  //cont ID,first name,last name, tribe
                  Contestant c1 = new Contestant();
                  c1.setID("i1");

                  Contestant c2 = new Contestant();
                  c2.setID("i2");

                  Contestant c3 = new Contestant();
                  c3.setID("i3");

                  Contestant c4 = new Contestant();
                  c4.setID("i4");

                  Contestant c5 = new Contestant();
                  c5.setID("i5");

                  Contestant c6 = new Contestant();
                  c6.setID("i6");


                  int response = addContestant(c1);
                  response+=addContestant(c2);
                  response+=addContestant(c3);
                  response+=addContestant(c4);
                  response+=addContestant(c5);
                  response+=addContestant(c6);

                  System.out.println("TEST 1: ADDING 6(MAX AMOUNT IN THIS CASE)
CONTESTANTS");
                  if(response!=0)
                        System.out.println("\tFAILED ADDING CONTESTANTS UP TO MAX
AMOUNT");
                  else
                        System.out.println("\tSUCCESFULLY ADDED CONTESTANTS UP TO
```

```
MAX AMOUNT");

                    System.out.println("\n\nTEST 2: ADDING 7( MORE THAN MAX AMOUNT IN
THIS CASE) CONTESTANT");
                    Contestant c7 = new Contestant();
                    c7.setID("i7");
                    response = addContestant(c7);
                    if(response==-2)
                           System.out.println("\tPASSED: COULD NOT ADD MORE THAN MAX
CONTESTANTS");
                    else
                           System.out.println("\tFAILED: COULD ADD MORE THAN MAX
CONTESTANTS");


                    System.out.println("\n\nREMOVING A CONTESTANT TO TEST FOR SAME
ID");
                    list.remove(5);
                    System.out.println("TEST 3: ADDING CONTESTANT WITH ID i5(which
already exists)");
                    Contestant c8 = new Contestant();
                    c8.setID("i5");
                    response = addContestant(c8);
                    if(response==-1)
                           System.out.println("\tPASSED: COULD NOT ADD CONTESTANT WITH
SAME ID");
                    else
                           System.out.println("\tFAILED: COULD ADD CONTESTANT WITH
SAME ID");



             } catch (InvalidFieldException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
             }

      }

      //STUB to add contestants RETURNS -1 IF ID INVALID, -2 IF MAX CONTESTANT
REACHED
      public static int addContestant(Contestant c){
             if(NUM_TEST_CONTESTANTS>list.size()){
                    if(isIDValid(c.getID()))
                           list.add(c);
                    else
                           return -1;
             }else{
                    return -2;
             }
             return 0;
      }

      //STUB to to see if id is a duplicate
      public static boolean isIDValid(String id){
             for(Contestant c:list){
                    if(c!=null&&id.equals(c.getID()))
                           return false;
             }
             return true;
      }
```

```
	public static void main(String[] args) throws InvalidFieldException {
		driver();

		stub();
	}
}
```