Carly Heibel

CSSE376

Lab 3

3/18/2014

1. The source code is found in the “Expedia” project and the tests are found in the “Expedia Test” project.
2. The project contains the following classes:
   1. Car
   2. Flight
   3. Hotel
   4. User
3. The Flight class implements the Booking interface and keeps track of the date that a flight leaves and the date that it returns. It also has a “Miles” field. At this point it only has methods to determine if two flights are the same. It also has a method called “getBasePrice” which uses a very simple algorithm for calculating the price based on how many days the flight is spread out over.
4. The test classes in the project are the following:
   1. BookingTest
   2. CarTest
   3. FlightTest
   4. HotelTest
   5. UserTest
5. The test methods in the UserTest class are the following:
   1. Setup
   2. TestThatUserInitializes
   3. TestThatUserHasZeroFrequentFlierMilesOnInit
   4. TestThatUserCanBookEverything
   5. TestThatUserHasFrequentFlierMilesAfterBooking
   6. TestThatUserCanBookAFlight
   7. TestThatUserCanBookAHotelAndACar
   8. TestThatUserHasCorrectNumberOfFrequentFlyerMilesAfterOneFlight
   9. TearDown
6. Below are some functions supported by the Assert class:
   1. AreEqual
   2. AreNotEqual
   3. AreNotSame
   4. AreSame
   5. Contains
7. Explanations
   1. AreEqual – verifies that two objects have the same value
   2. AreNotEqual – verifies that two objects do not have the same value
   3. AreNotSame – Verifies that two object references do not refer to the same object
   4. AreSame – verifies that two object references refer to the same object
   5. Contains – asserts that an object is contained in a list
8. The difference is that AreEqual and AreNotEqual check the *values*; AreSame and AreNotSame check the *references/addresses*.
9. The unit test “TestThatHotelInitializes” verifies that the constructor actually creates a Hotel object.
10. The generic algorithm for calculating “getBasePrice” is to multiply the number of nights by $45.
11. The cases where only one night is booked, where two nights are booked, and where 10 nights are booked.
12. Since we already verified that the constructor works, prior to executing these test cases, there is no need to repeat that process in each of our tests (which are meant not to test whether the constructor works but rather that another function does what it is supposed to).
13. This test case expects an exception because the constructor for “Hotel” verifies that the parameter is greater than zero and throws an ArgumentOutOfRangeException if it is not.
14. [ExpectedException(typeof(OutOfMemoryException))]