Trevor Krenz

Lab 2

3/13/12

1. The project source code is found in Expedia. The test source code is found in ExpediaTest.
2. The classes in the project are: AssemblyInfo, Booking, Car, Flight, Hotel, User, BookingTest, CarTest, FlightTest, HotelTest, and UserTest.
3. The each instance of the Flight class keeps track of a particular flight's leave time, return time, and the miles of the flight.
4. The test classes are BookingTest, CarTest, FlightTest, HotelTest, and UserTest.
5. The test methods in UserTest are TestThatUserInitializes, TestThatUserHasZeroFrequentFlierMilesOnInit, TestThatUserCanBookEverything, TestThatUserHasFrequentFlierMilesAfterBooking, TestThatUserCanBookAFlight, TestThatUserCanBookAHotelAndACar, and TestThatUserHasCorrectNumberOfFrequentFlyerMilesAfterOneFlight.
6. AreEqual, AreNotEqual, AreNotSame
7. AreEqual verifies the given values are equal, else throw an exception. AreNotEqual verifies the functions are not equal, else throw an exception. AreNotSame verifies that two objects do not refer to the same object, else throw an exception.
8. AreEqual checks if the two values given have the same values. This is equalvalent to the 'equals' method in Java. AreSame checks if two objects have the same reference. This is equivalent to '==' in Java.
9. The unit test is verifying that making a Hotel object does not instead return a null pointer.
10. It returns 45 multiplied by the number of nights to rent (stored as a field variable).
11. TestThatHotelHasCorrectBasePriceForOneDayStay() verifies that one day will cost 45. TestThatHotelHasCorrectBasePriceForTwoDayStay() verifies that two days will cost 90. TestThatHotelHasCorrectBasePriceForTenDaysStay() verifies that 10 days will cost 450.
12. That functionality has already been tested in another test case, TestThatHotelInitializes().
13. If the number of nights to rent is 0 or less, it is supposed to throw an ArgumentOutOfRangeException. Since -5 <= 0, an exception should be thrown for the test.
14. [ExpectedException(typeof(OutOfMemoryException))]