Ian Cundiff

Kyle Apple

Team14

1) The source code is in the Expedia project and the test code is in the ExpediaTest project.
2) AssemblyInfo, Booking, Car, Flight, Hotel and User.
3) It keeps track of the date the flight leaves, returns and miles travelled. It also allows for seeing whether two flights are the same based on this information.
4) BookingTest, CarTest, FlightTest, HotelTest and UserTest.
5) TestThatUserInitializes, TestThatUserHasZeroFrequentFlyerMilesOnIt, TestThatUserCanBookEverything, TestThatUserHasFrequentFlyerMilesAfterBooking, TestThatUserCanBookAFlight, TestThatUserCanBookAHotelAndACar and TestThatUSerHasACorrectNumberOfFrequentFlyerMilesAfterOneFlight.
6) AreEqual, AreNotEqual and AreNotSame.
7) AreEqual tests if two objects are equal by value, AreNotEqual tests if two objects are NOT equal by value, AreNotSame tests to see if two objects do NOT point to the same object (not the same memory references).
8) AreEqual tests for equality by determining if two objects have the same values, while AreSame tests by whether they actually are the same object (ie point to the same place).
9) That the Hotel constructor actually works when you pass a value in to it. It does this by making a new hotel object with a non-null value and then testing to make sure it is not null.
10) It multiplies the price ($45) by the number of nights stayed.
11) 1 day stay, 2 day stay and 10 day stay.
12) We are just testing the base price calculations, no the constructor. If we tested the constructor here too it wouldn’t be a unit test anymore.
13) Nights to rent must be greater than 0, so we want the program to throw an exception if it gets a negative number of days.
14) [ExpectedException(typeof(OutOfMemoryException))]