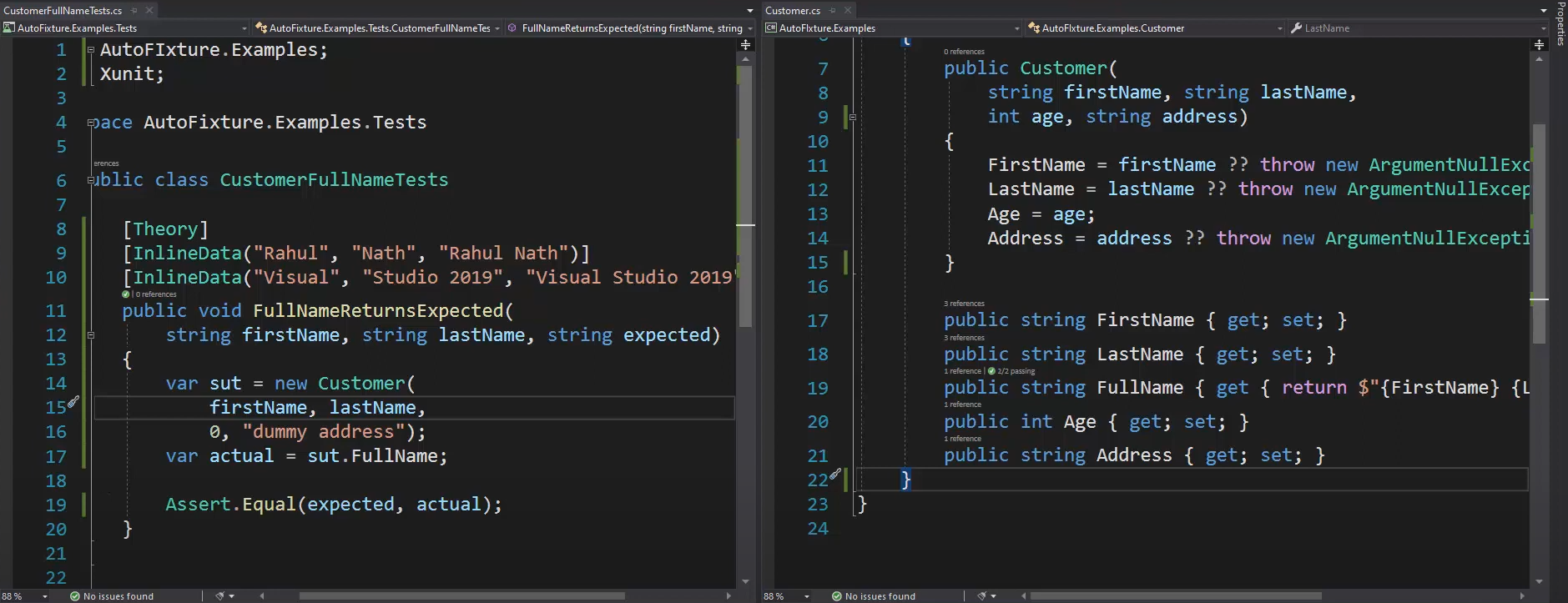
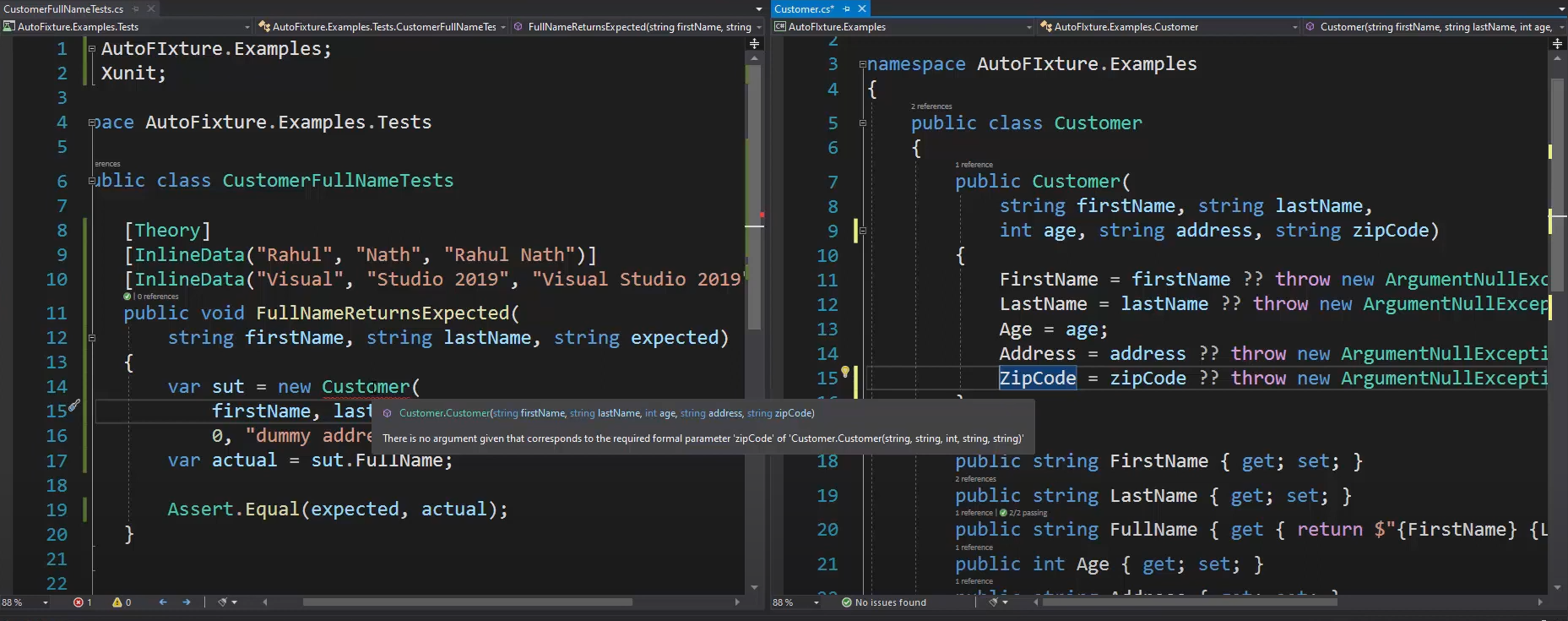
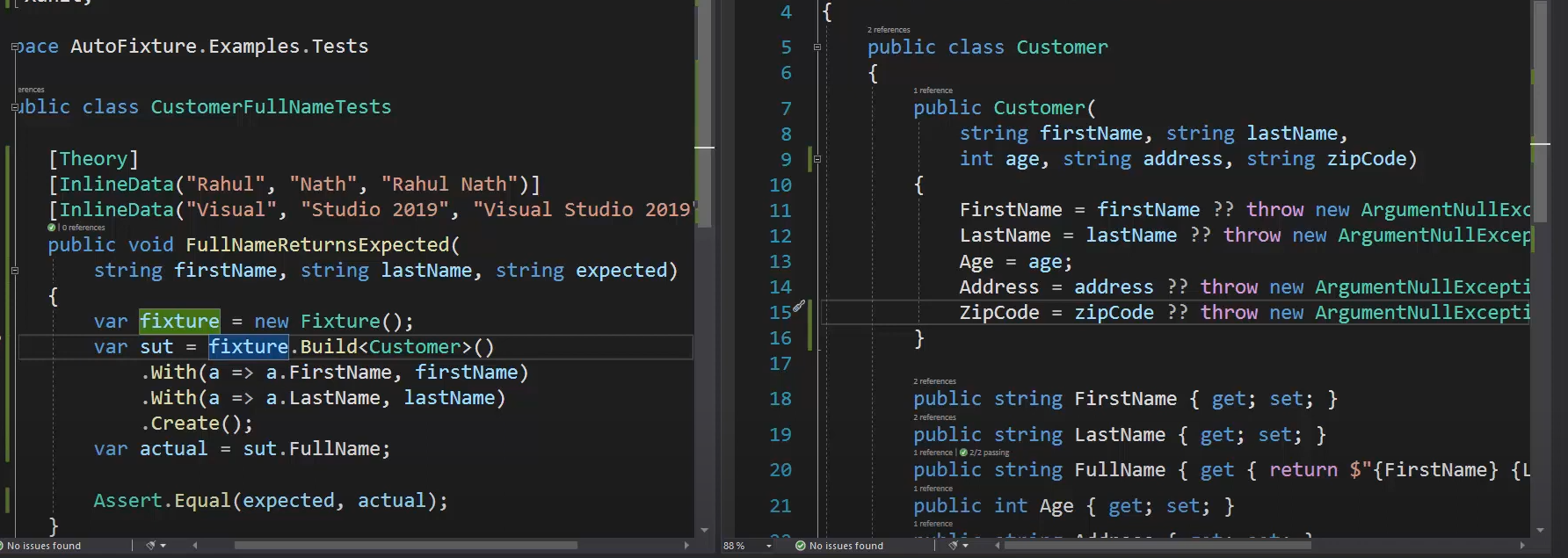
Normal Test case without Autofixture, just plain inline data and assert (XUnit)

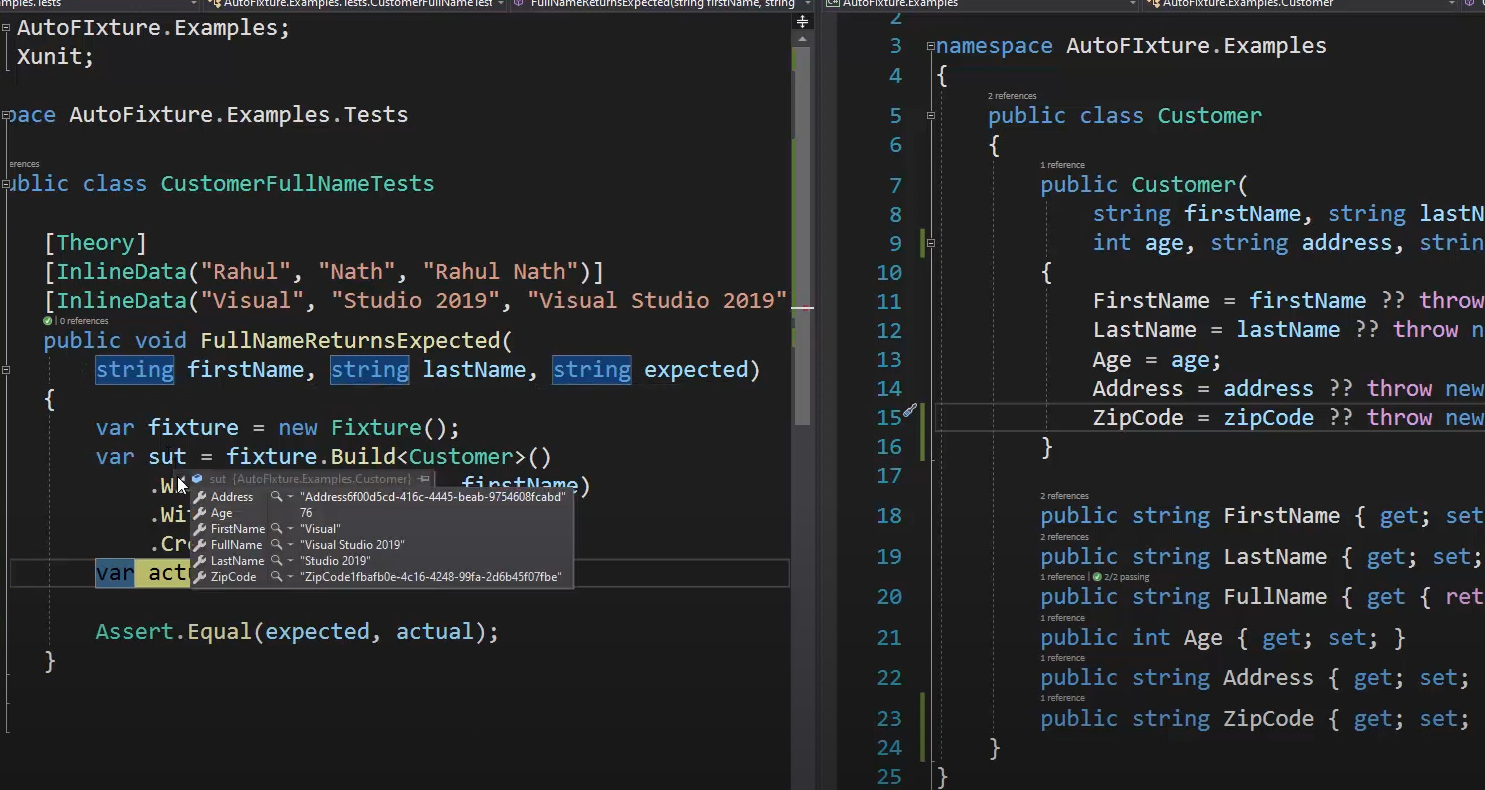


If a new property is added to the Customer class say zipcode and corresponding constructor, then we have to update the new-up in the test case as well



So we use Autofixture to create the class for us, it will generate random data for the properties that we do not specify, otherwise it will use the one that we provide.

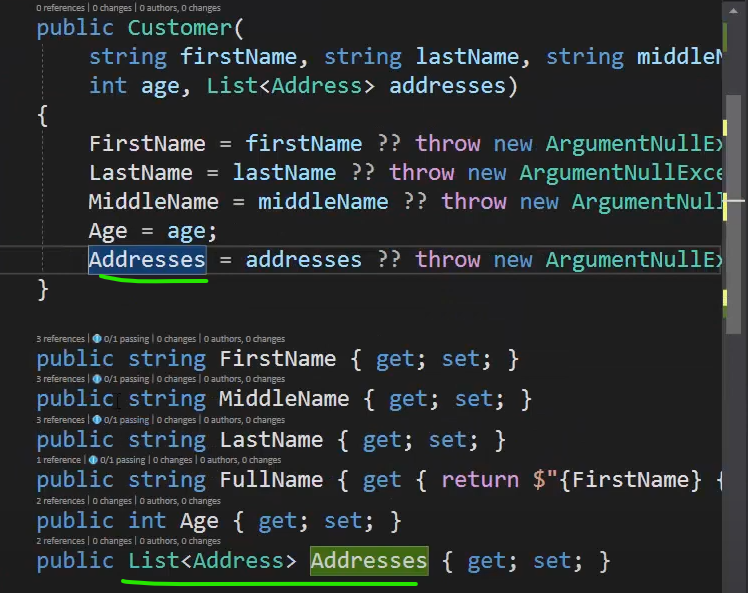


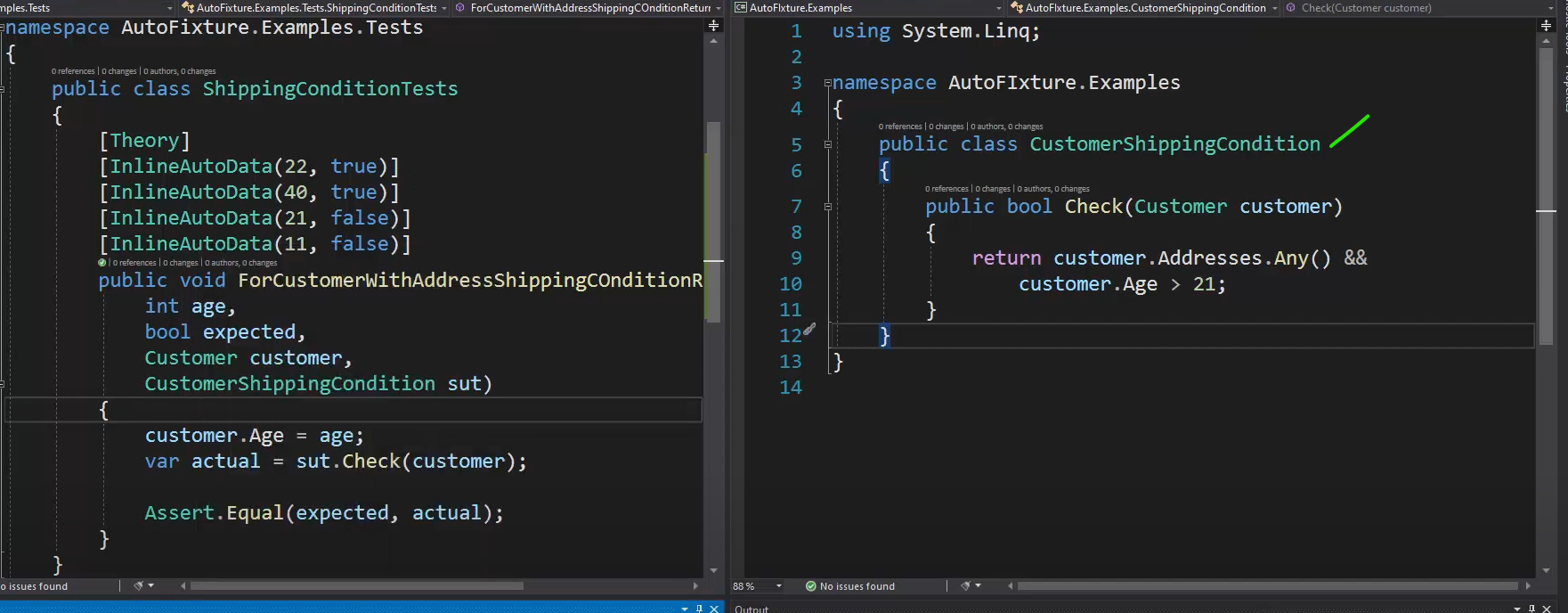


We can further cleanup the code inside the testcase by passing InlineAutoData and setting the fields that we care for.

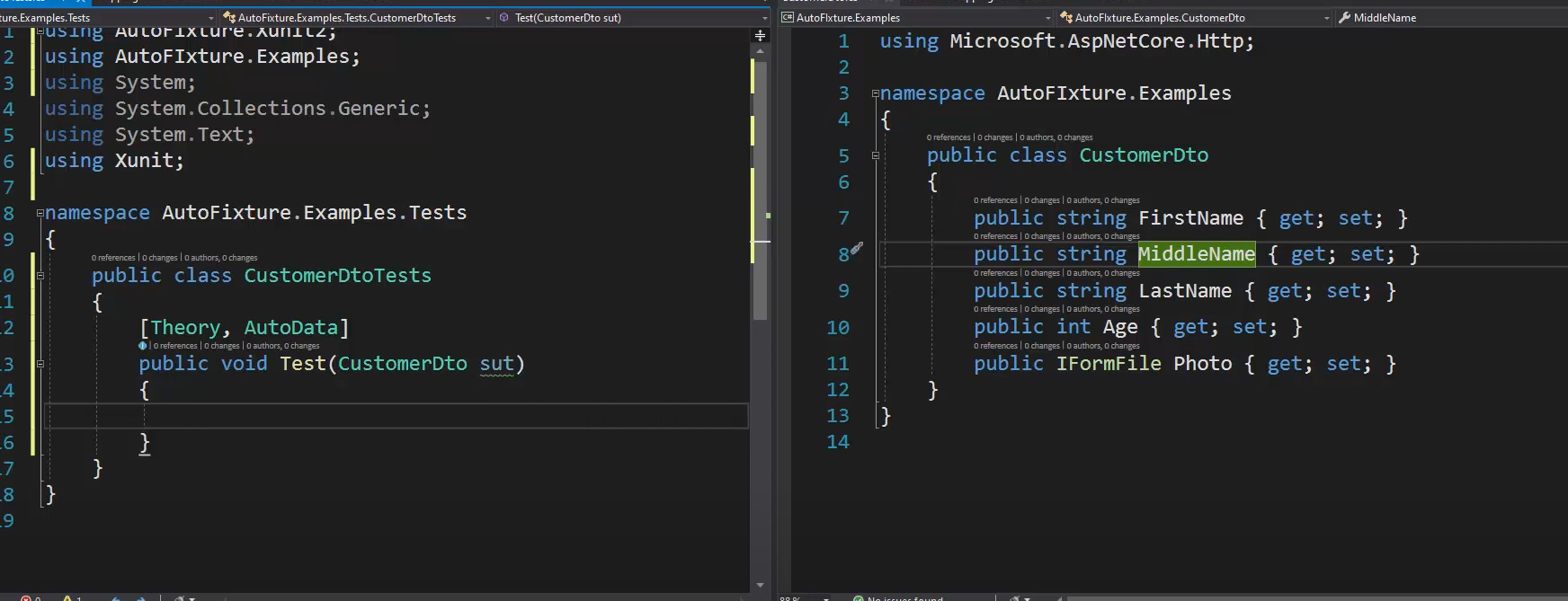


Say we have updated the Address with a List<Address> and new condition for Shipping:



Autofixture populates the list with count 3 by default behaviour.

Another example, if we have interface in class definition:

Then fixture will fail to instantiate