Software Project / OS
software-engineering

Egg 16-Mar-20 05:08 PM
Chat Box, for Software Engineering: Lab 2 Part 4

JohnnyH 16-Mar-20 05:13 PM
So the point of Part 4 is to create 2 new test case for our program

Edddddddy 16-Mar-20 05:14 PM
Yes

Sam 16-Mar-20 05:14 PM
Yep

JohnnyH 16-Mar-20 05:14 PM
Let us brainstorm a bit on what kind of test cases we should implement
I believe this was Question#4 of Part 3, what did you guys put for that question?

Sam 16-Mar-20 05:15 PM
I was having trouble thinking up some good one so I tried making a new function like printing out the products and then the test could be to check whether we have that product (edited)

JohnnyH 16-Mar-20 05:16 PM
That was mine too. I named it CheckInStock/CheckProduct

Edddddddy 16-Mar-20 05:16 PM
My idea was to implement a feature to remove an item from the list of products and then test to see if it was removed in a test case
The CheckInStock function could check the qnt value of the product and return true of false if empty
I guess we could make a those two functions

Egg 16-Mar-20 06:03 PM
Those are all very good ideas. Lets try implementing them test cases soon. (edited)

Eddddddddy 16-Mar-20 06:05 PM
I guess since Sam and Jinquan both were interested in the CheckInStock feature they could implement that and Tommy and I will implement the Remove feature

Egg 16-Mar-20 06:05 PM
Good idea! Is there a time where we can all meet before Wednesday 11:59pm? (edited)

Sam 16-Mar-20 06:06 PM
Got it I'll coordinate with Jingquan

Eddddddddy 16-Mar-20 06:08 PM
Yea we can meet right before class if that's possible to see what we got

Sam 18-Mar-20 01:54 PM
Hey Jinquan so I got a question about that first line where we're trying to access the quantity in the dictionary(I believe thats what we call it in python)

```
if item. ["qnt"] > 0:
```

I've never seen a code where you access it using a "."

JohnnyH 18-Mar-20 01:54 PM
of course, whats your question?
ah, my bad. I was still thinking in java
Lets change that

Sam 18-Mar-20 01:55 PM
The way I've done mine would be either item["qnt"] or using a "if item in product" (edited)
sort of way
ok

JohnnyH 18-Mar-20 01:56 PM
yeah that'll work

Sam 18-Mar-20 02:01 PM
wait jinquan does the code even work?

```
#TODO: Sam and Jinquan: Implement CheckInStock Function
def checkInStock(self, products, item):
    if item["qnt"] > 0:
        print("Product In Stock")
    else:
        print("Product Out of Stock")
```

im pretty much a noob at python

JohnnyH 18-Mar-20 02:01 PM
same here man

Sam 18-Mar-20 02:02 PM
im typing it into pycharm and its not working

JohnnyH 18-Mar-20 02:02 PM
that was just the preliminary draft

Sam 18-Mar-20 02:02 PM
oh ok

JohnnyH 18-Mar-20 02:02 PM

```
def checkInStock(self, products):
    for k, v in products.items():
        if self.items['qnt'] == products:
            if products['qnt'] > 0:
                return True
    return False
```

Thats my updated version

but I'm getting this error

```
    def checkInStock(self, products):
        for k, v in products.items():
>           if self.items['qnt'] == products:
E       KeyError: 'qnt'

Invoice.py:51: KeyError
```

Sam 18-Mar-20 02:03 PM
yeah

so

i did it this way

```
def checkStock(self, products, product):
    if product in products:
        return True
    else:
        return False
```

doesn't actually check the quantity
actually this more or less searches if product is in the inventory or dictionary in code

---

JohnnyH 18-Mar-20 02:04 PM
oh I was looking for a simple 'contains' function for python
so all you have to do is type in

Sam 18-Mar-20 02:13 PM
Jinquan could you take a look at my code I wrote and make sure we're on the same page?

JohnnyH 18-Mar-20 02:29 PM
It's a fantastic start, we're halfway done
lets try to add the quantity check for when it return True

Sam 18-Mar-20 02:29 PM
got it

JohnnyH 18-Mar-20 03:04 PM
Seems like this is all we needed

```
#TODO: Sam and Jinquan: Implement CheckInStock Function
def checkInStock(self, products, item):
    for i, v in products.items():
        if i == item:
            if v['qnt'] > 0:
                return True
        else:
            return False
```

Sam 18-Mar-20 03:05 PM
ok nice
I'm getting this error and I essentially have what you have

```
    for i, v in products.items:
TypeError: 'builtin_function_or_method' object is not iterable
```

Oh

JohnnyH 18-Mar-20 03:10 PM
you need it () im guessing?

Sam 18-Mar-20 03:10 PM
Ok I see whats wrong now I got it working
thanks
Ok I was able to get my test cases working how's it going on your end?

JohnnyH 18-Mar-20 03:14 PM
I got mine to work too
I believe we are done with our test case now

Sam 18-Mar-20 03:14 PM
ok great

Sam 18-Mar-20 03:22 PM
I added a couple more lines to the code if you could take a look at it

JohnnyH 18-Mar-20 03:24 PM
That's great. It looks even better

Sam 18-Mar-20 03:24 PM
We should try to have 1 or 2 more test cases just to make sure it works

Eddddddddy 18-Mar-20 04:35 PM
Ok Tommy we should start working on our side of the project

Egg 18-Mar-20 04:35 PM
ok no problem.

Eddddddddy 18-Mar-20 04:35 PM
What do you think the overall flow of the function show look like

Egg 18-Mar-20 04:36 PM
Hmmm. Good question, im not too well versed on python so i'll take your suggestion.

Eddddddddy 18-Mar-20 04:37 PM
I think because we are going to pass in a dictionary of products to the method and the specific item we want to remove that it should return a modified version of that products dictionary

Egg 18-Mar-20 04:37 PM
That sounds about right. Sounds logical on my end.

Edddddddy 18-Mar-20 04:37 PM

```
    def removeItem(self, products, item):
        return modifiedProducts
```

so kinda like this where modifiedProducts is a dict
I'll work on the code and you can work on the test case how about that?

Egg 18-Mar-20 04:38 PM
no problem. i'll get started right away
i'll let you know when i'm done with the test cases

Egg 18-Mar-20 04:46 PM
Im going to make some fixtures to test the pen item and then add a water bottle item to the list of products and test that as welll

Edddddddy 18-Mar-20 04:47 PM
Ok so it seems like this isn't too difficult, after some googling I found that you can just copy over a dictionary using a list comprehension and then pop the item that you don't want
that sounds good then

Egg 18-Mar-20 04:49 PM
That's good. Here are the fixtures. Do they seem ok?

```
@pytest.fixture()
def item_Pen():
    item_Pen = "Pen"
    return item_Pen


@pytest.fixture()
def item_Water():
    item_Water = "Water Bottle"
    return item_Water
```

Edddddddy 18-Mar-20 04:49 PM
Yes they seem good
just make sure you add a water bottle item to the list of products at the top of the testInvoice file

```
def removeItem(self, products, item):
    modifiedProducts = {i: v for i, v in products.items()}
    modifiedProducts.pop(item)
    return modifiedProducts
```

this is what my code looks like rn
basically the function is going to return the whole dictionary without the specific element in it
so assert that when you test (edited)

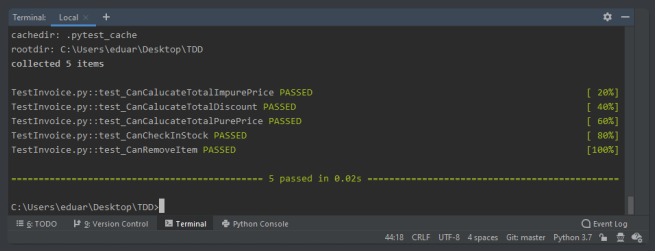
Egg 18-Mar-20 04:58 PM
ok.
Hey, can you check if this looks about right?

```
def test_CanRemoveItem(invoice, products, item_Pen, item_Water):
    invoice.removeItem(products, item_Pen)
    invoice.removeItem(products, item_Water)
    assert invoice.removeItem(products, item_Pen) == {'Notebook': {'qnt': 5, 'unit_price': 7.5, 'discount': 10},
                                                      'Water Bottle': {'qnt': 0, 'unit_price': 1.0, 'discount': 7}}
    assert invoice.removeItem(products, item_Water) == {"Pen": {"qnt": 10, "unit_price": 3.75, "discount": 5},
                                                        "Notebook": {"qnt": 5, "unit_price": 7.5, "discount": 10}}
```

Edddddddy 18-Mar-20 05:04 PM
Mmmm
Yea thats perfect (edited)
Just put it in the repl and I'll copy it over and test it

Egg 18-Mar-20 05:06 PM
ok. Sounds good to me (edited)

Edddddddy 18-Mar-20 06:03 PM
everything works



Exported 95 message(s)