Richard Hayes Crowley

02/18/2021

CSC\_157\_Lab\_010

**SOURCE CODE:**

*# Richard Hayes Crowley*

*# 03/18/2020*

*# CSC\_157\_LAB\_010*

*import* json

*from* os *import* path

*from* pprint *import* pprint, pformat

class AncillaryException(Exception):

"""Custom exception for not entering 'y' or 'n' for ancillary CD ROM"""

*pass*

class Inventory(object):

*# changing the structure of my class variable from a list to a dict, preload with previous JSON state if it exists*

*# structure will use titles as keys, since titles must be unique, and title.author, title.description, title.ancillary as properties*

inventory = json.load(open("inventory.json", "r")) *if* path.exists(

"inventory.json") *else* {}

def \_\_init\_\_(*self*, *title*, *description*):

*# setting title as key for another dict which will hold description as a key/value pair*

Inventory.inventory[title] = {"description": description}

@staticmethod

def inventoryCheck(*title*):

*return* title in Inventory.inventory

*# check if the title already exists in the list of titles*

class Books(Inventory):

def \_\_init\_\_(*self*, *title*, *description*, *author*, *ancillary*=None):

*# execute the parent constructor*

super().\_\_init\_\_(title, description)

*# setting author and ancillary for title in inventory*

Books.inventory[title]["author"] = author

Books.inventory[title]["ancillary"] = True *if* ancillary == "y" *else* False

def get\_inventory():

*return* Books.inventory

def \_\_str\_\_(*self*):

current\_stock = json.load(open("inventory.json", "r")) *if* path.exists(

"inventory.json") *else* self.inventory

*# format dict to prettified string*

*return* pformat(current\_stock)

oscar = Books("An Ideal Husband",

"Wilde's scintillating drawing-room comedy ", "Oscar Wilde")

title\_to\_check = input("Title to check in inventory: ")

*if* (Books.inventoryCheck(title\_to\_check)): *# perform a check*

print("The title already exists; it cannot be added to the inventory.")

*else*:

print('"' + title\_to\_check + '" does not exist in inventory.')

print('Enter information about "' + title\_to\_check + '":\n')

author = input("Author of book: ")

description = input("Description of book: ")

*while* True:

*try*:

ancillary = input(

"Does this book come with a CD-ROM? Enter y or n: ")

*if* ancillary != "y" and ancillary != "n":

*raise* AncillaryException

*except* AncillaryException:

print("Please enter either 'y' or 'n'")

*else*:

*break*

updated\_inventory = Books(title\_to\_check, description, author,

ancillary *if* ancillary == "y" *else* None)

*# add the updated inventory book to json*

json.dump(Books.get\_inventory(), open("inventory.json", "w"),

*indent*=4, *sort\_keys*=True)

print('\n"' + title\_to\_check + '" has been added to the inventory.\n')

print(f"Inventory to date is: {updated\_inventory}")

**OUTPUT:**Text

Description automatically generated

**DATA PERSEVERANCE (INVENTORY.JSON SECONDARY STORAGE):**

{

"An Ideal Husband": {

"ancillary": false,

"author": "Oscar Wilde",

"description": "Wilde's scintillating drawing-room comedy "

},

"Crime & Punishment": {

"ancillary": false,

"author": "Dostoevsky",

"description": "the truth will out"

},

"Great Expectations": {

"ancillary": true,

"author": "Charles Dickens",

"description": "High hopes dashed."

}

}