Celeste Hernandez Mora

Professor Nguyen

8 February 2024

CIS-30A-21273

Course Project Part 1: Documenting and Planning Project

The primary goal of the program is to demonstrate a business transaction through the usages of multiple Python features, such as but not limited to modules, functions, control statements, exceptions, classes, and text file modifications. It is functional in that it successfully writes a text file and outputs it to the user with correct information, as submitted by the user. The intended audience for this program would be people who are looking for a demonstration of a business transaction, recorded in a file for them and then output. Though the program is functional, there are still some features it lacks. Its strength lies in the fact it can accurately write a text file with the user’s selected goods and appointments, though there is a weakness in that there is no graphical user interface implemented into the program for easier readability. From that, one significant improvement that could be made to the program in the future would be the implementation of a graphical user interface. It would allow for easier navigation through the menus of the program. As for the code of the program, it is mostly clean that functions are sorted according to their purpose, especially under classes. All functions and classes then come together to create the text file containing the goods and appointment as selected by the user. Overall, the program should be readable in all compartments, from modules to text files to output.

**See Pseudocode Below:**

The program's purpose is to conduct a business transaction according to the user's selections.

MODULE main.py

Import shopping

Function executeShop

Pass In: shop

call: shop.displayMenu()

call: shop.addItem()

Pass Out: none

Endfunction

Function executeDeliver

Pass In: deliver

call: deliver.displayMenu()

call: deliver.selection()

GET choice

Pass Out: choice

Endfunction

PRINT introduction and prompt

INITIALIZE purchases

INITIALIZE sale

call: shopping.PurchaseItem(sale, purchases)

GET shop

PRINT delivery prompt

INITIALIZE dates

call: shopping.DeliverItem(dates)

GET deliver

call: executeDeliver(deliver)

GET choice

call: shopping.InfoPrint(sale, purchases, dates, choice)

INITIALIZE show

call: show.combine()

END main.py

MODULE shopping.py

Class: PurchaseItem

Pass In: None

Function: \_\_init\_\_(self, sale, purchases):

Pass In: self, sale, purchase

GET self.sale

GET self.purchases

Pass Out: None

Endfunction

Function: displayMenu(self):

Pass In: self

FOR each item in sale

IF item != 6 THEN

PRINT item

ELSE

PRINT exit

ENDIF

ENDFOR

Pass Out: None

Endfunction

Function: addItem(self):

Pass In: self

INITIALIZE total

INITIALIZE choice

WHILE choice is not exit number

BEGIN assertion

GET choice

EXCEPTION AssertionError

WHEN choice is not <= 6

PRINT error

ELSE

IF choice != 6

PRINT selection

ELSE

PRINT exit

BREAK

ENDIF

END

ENDWHILE

APPEND choice to purchases

call: currentCart(total)

SET total to 0.00

Pass Out: None

Endfunction

Function: currentCart(self, total)

Pass In: None

PRINT items

FOR each item in self.purchases

PRINT item

CALCULATE total

ENDFOR

PRINT total

Pass Out: None

Endfunction

Pass Out: None

Endclass

Class: DeliverItem

Pass In: None

Function: \_\_init\_\_(self, dates)

Pass In: self, dates

INITIALIZE self.dates

Pass Out: None

Endfunction

Function: displayMenu(self)

Pass In: self

FOR each item in self.dates

PRINT item

ENDFOR

Pass Out: None

Endfunction

Function: selection(self)

Pass In: self

BEGIN assertion

GET choice2

EXCEPT AssertioNError

WHEN choice2 > 6

PRINT error

ELSE

PRINT delivery date

Pass Out: choice2

Endfunction

Pass Out: None

Endclass

Class: InfoPrint(PurchaseItem)

Pass In: PurchaseItem

Function: \_\_init\_\_(self, sale, purchases, dates, choice)

Pass In: self, sale, purchasees, dates, choice

INHERIT PurchaseItem.\_\_init\_\_(self, sale, purchases)

GET self.dates

GET self.choice

Pass Out: None

Endfunction

Function: display(self)

Pass In: self

OPEN purchase.txt

WRITE text

CLOSE

Pass Out: None

Endfunction

Function: displayItems(self)

OPEN purchase.txt

APPEND items

FOR each item in self.purchases

APPEND item

ENDFOR

CLOSE

Pass Out: None

Endfunction

Function: displayTotal(self)

Pass In: self

INITIALIZE total

FOR each item in self.purchases

CALCULATE total

ENDFOR

OPEN purchase.txt

APPEND total

CLOSE

Pass Out: None

EndFunction

Function: displayDate(self)

Pass In: self

OPEN purchase.txt

APPEND delivery date

Pass Out: None

Endfunction

Function: combine(self)

Pass In: self

call: self.display()

call: self.displayItems()

call: self.displayTotal()

call: self.displayDate()

OPEN purchase.txt

READ line

FOR each line

PRINT line

ENDFOR

CLOSE

Pass Out: None

Pass Out: None

Endclass

END shopping.py