'''

Keni Mou

Kmou1@binghamton.edu

Assignment 8, Exercise 1

Lab Section 52

CA Kyle Mille

'''

'''

Anylize

Output to the monitor:

items(str)-name

count(int)

average(float)

Tasks:

tupleListToDict:convert list to dictionary

getSortedKeyList: Compute the average score

computeAverage: Compute the average score

getSortedListOfTuples: sort the keys from a list

'''

#convert list to dictionary

#param theList(list)

#param dic(dict) -the dictionary

#param valueList(list) - contain the values

#return dic

def tupleListToDict(theList):

dic={} #the dictionary

for i in theList:

if i[0] not in dic:

if len(i)>1:

dic[i[0]]=[i[1]]

else:

dic[i[0]]=[]

else:

if len(i)>1:

valueList=dic[i[0]]

valueList.append(i[1])

dic[i[0]]=valueList

return dic

#sort the keys from a dic

#param dic(dict)

#param keysList(list)

#param sortedDic(dict)

# return sortedDic

def getSortedKeyList(dic):

keysList=list(dic.keys())

## print(keysList) #debug

keysList.sort()

return keysList

#Compute the average score

#param theList(list)

#param theSum(float)

#param counter(int)

#param ave(float)

#return ave

def computeAverage(theList):

theSum=0

counter=0

for i in theList:

if i:

theSum += i

if i:

counter += 1

if theSum !=0:

ave=theSum/counter

else:

ave=0

return ave

#sort the keys from a list

#param tupleList(list)

#return tupleList

def getSortedListOfTuples(tupleList):

tupleList.sort()

return tupleList

def main():

gradeList= [ ('Zaphod', 33), ('Zaphod', 75), ('Slartibartfast', ),

('Trillian', 98), ('Trillian', 97), ('Slartibartfast', ),

('Marvin', 500) , ('Arthur', 20), ('Arthur', 64),

('Trillian', 99), ('Marvin', 450), ('Marvin', 550),

('Agrajag', 85), ('Agrajag', ), ('Agrajag', ),

('Ford', ), ('Ford', ), ('Ford', 50) ]

#the first set---------------------------------

gradeDict= tupleListToDict(gradeList)

sortedList=getSortedKeyList(gradeDict)

#display

print("%30s"%"Grade")

print("%16s"%"Name",'\t',"%2s"%"Count",'\t', "%10s"%"Average")

print("---------------------------------------------")

for items in sortedList:

average=computeAverage(gradeDict[items])

count=len(gradeDict[items])

print("%16s"%(items),'\t',"%2d"%(count),'\t', "%10.2f"%(average))

#the second set----------------------------------

tupleList=list(gradeDict.items())

## print(tupleList) #debug

sortedKeyWords=getSortedListOfTuples(tupleList)

#display

print("%30s"%"Grade")

print("%16s"%"Name",'\t',"%2s"%"Count",'\t', "%10s"%"Average")

print("---------------------------------------------")

for items in sortedKeyWords:

average=computeAverage(items[1])

## print (average) #debug

count=len(items[1])

## print(count) #debug

print ("%16s"%(items[0]),'\t',"%2d"%(count),'\t', "%10.2f"%(average))

main()

Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>> ================================ RESTART ================================

>>>

Grade

Name Count Average

---------------------------------------------

Agrajag 1 85.00

Arthur 2 42.00

Ford 1 50.00

Marvin 3 500.00

Slartibartfast 0 0.00

Trillian 3 98.00

Zaphod 2 54.00

Grade

Name Count Average

---------------------------------------------

Agrajag 1 85.00

Arthur 2 42.00

Ford 1 50.00

Marvin 3 500.00

Slartibartfast 0 0.00

Trillian 3 98.00

Zaphod 2 54.00

>>>