Richard Hayes Crowley

03/26/2021

CSC\_157\_Lab\_017

**SOURCE CODE (2 files):**

*from* directoryMethods *import* \*

def main():

*while* True:

menu()

choice = int(input("Enter Menu Choice Now! "))

*if* (choice == 1):

empFile = open("employees.txt", "r")

*for* line *in* empFile:

tokens = line.split(" ")

print(tokens[0], "", tokens[1], "",

(float(tokens[2]) \* float(tokens[3])))

*if* (choice == 2):

name = input("Enter a name to search an employee ")

emp = findEmployee(name)

*if* not emp:

print("employee not found!")

*else*:

print(f"{emp[1][0]} {emp[1][1]}")

*if* (choice == 3):

createEmployee()

*if* (choice == 4):

name = input(

"Enter the name of the employee you want to delete (First Last): ")

deleteEmployee(name)

*if* (choice == 5):

name = input(

"Enter the name of the employee you want to update (First Last): ")

updateEmployee(name)

*if* (choice == 6):

print("\nGoodbye!")

*break*

exit()

*if* \_\_name\_\_ == "\_\_main\_\_":

main()

**directoryMethods.py**

def menu():

pstr = "\nChoose from the following payroll choices\n"

pstr += "(1) A gross PR payroll report for all employees\n"

pstr += "(2) A gross PR payroll report for a single employee by name\n"

pstr += "(3) Enter new employee data\n"

pstr += "(4) Delete employee from directory\n"

pstr += "(5) Update employee in directory\n"

pstr += "(6) Quit Program"

print(pstr)

def createEmployee():

*while* True:

*try*:

firstName = input("\nPlease enter employee's first name: ").strip()

lastName = input("\nPlease enter employee's last name: ").strip()

rate = float(input("\nEnter hourly rate: ").strip())

hoursWorked = float(input("\nEnter hours worked: ").strip())

*break*

*except* ValueError:

print("Please enter a number for rate and hours worked.")

empStr = f"\n{firstName} {lastName} {rate} {hoursWorked}"

empFile = open("employees.txt", "a")

empFile.write(empStr)

print(f"\n{empStr} added to employee directory.")

def findEmployee(*name*):

empFile = open("employees.txt", "r")

line = empFile.readline()

flag = False

*while*(line != ""):

tokens = line.split(" ")

empName = tokens[0] + " " + tokens[1]

*if* (name == empName):

*return*((line,), (empName, "$%4.2f" %

(float(tokens[2]) \* float(tokens[3]))))

*break*

*else*:

flag = True

line = empFile.readline()

*if* (flag == True):

*return* False

def deleteEmployee(*name*):

employee = findEmployee(name)

*if* not employee:

print("Employee not found!")

*else*:

*with* open("employees.txt", "r") *as* f:

lines = f.readlines()

*with* open("employees.txt", "w") *as* f:

*for* line *in* lines:

tokens = line.split(" ")

empName = tokens[0] + " " + tokens[1]

*if* empName.lower() != name.lower():

f.write(line)

*else*:

print(f"\n{empName} removed from directory.")

def updateEmployee(*name*):

employee = findEmployee(name)

*if* not employee:

print("Employee not found!")

*else*:

employeeLine = employee[0][0]

*with* open("employees.txt", "r") *as* f:

lines = f.readlines()

newLines = []

*for* line *in* lines:

*if* line == employeeLine:

print(f"\nCurrent data for this employee: {line}")

*while* True:

*try*:

choice = input(

f"Which would you like to change?\n 1. Rate \n2. Hours Worked\nChoice: ").strip()

*if* not choice or choice not in ["1", "2"]:

*raise* exception

*break*

*except*:

print("Please choose 1 or 2")

*if* choice == "1":

*while* True:

*try*:

updatedRate = float(

input("\nPlease enter an updated rate: "))

*break*

*except* ValueError:

print("\nPlease enter a number!")

split\_line = line.split(" ")

split\_line[2] = str(updatedRate).strip()

newLine = " ".join(split\_line)

newLines.append(newLine)

*else*:

*while* True:

*try*:

updatedHours = float(

input("\nPlease enter updated hours: "))

*break*

*except* ValueError:

print("\nPlease enter a number!")

split\_line = line.split(" ")

split\_line[3] = str(updatedHours).strip()

newLine = " ".join(split\_line)

newLines.append(newLine)

*else*:

newLines.append(line)

*with* open("employees.txt", "w") *as* file:

file.writelines(newLines)

print("\nUpdated directory:\n")

[print(line) *for* line *in* newLines]

**OUTPUT ON NEXT PAGE (3 screenshots)**

**OUTPUT 1:**

Text

Description automatically generated

**OUTPUT 2:**

**Text

Description automatically generated**

**OUTPUT 3:**

**Text

Description automatically generated**

**EMPLOYEES.TXT SNAPSHOT (after all steps were completed):  
  
Graphical user interface, text, application

Description automatically generated**