Deborah Barndt

ITMD 513 Open Source Programming

Professor Dr. Sam

Hw10

4-8-19

'''

Deborah Barndt

4-8-19

Employee.py

hw10: Inheritance

This program will contain two classes: Employee and ProductionWorker. The Employee class

will keep data attributes for the following pieces of information: Employee name and

Employee number. The ProductionWorker will be a subclass of the Employee class. The

ProductionWorker class will keep data attributes for the following information: Shift

number (an integer, such as 1, 2, or 3) and Hourly pay rate.

The workday is divided into two shifts: day and night. The shift attribute will hold an

integer value representing the shift that the employee works. The day shift is shift 1 and

the night shift is shift 2. Write the appropriate accessor and mutator methods for each

class. Once you have written the classes, write a program that creates an object of the

ProductionWorker class and prompts the user to enter data for each of the object's data

attributes. Store the data in the object, then use the object's accessor methods to

retrieve it and display it on the screen.

Written by Deborah Barndt.

'''

# Class Employee that contains data attributes for Employee name and Employee number.

class Employee():

# Function to create a constructor of class Employee.

def \_\_init\_\_(self, employee\_name, employee\_num):

self.employee\_name = employee\_name

self.employee\_num = employee\_num

# Getter function to get employee name.

def getEmployeeName(self):

return self.employee\_name

# Setter function to set the employee name.

def setEmployeeName(self, employee\_name):

self.employee\_name = employee\_name

# Getter function to get employee number.

def getEmployeeNum(self):

return self.employee\_num

# Setter function to set the employee number.

def setEmployeeNum(self, employee\_num):

self.employee\_num = employee\_num

# Class ProductionWorker that contains the data attributes for Shift number and Hourly pay rate.

class ProductionWorker(Employee):

# Function to create a constructor of the class ProductionWorker.

def \_\_init\_\_(self, employee\_name, employee\_num, shift\_num, hourly\_rate):

# Invoke the constructor of Employee class.

Employee.\_\_init\_\_(self, employee\_name, employee\_num)

self.shift\_num = shift\_num

self.hourly\_rate = hourly\_rate

# Getter function to get shift number.

def getShiftNum(self):

return self.shift\_num

# Setter function to set the shift number.

def setShiftNum(self, shift\_num):

self.shift\_num = shift\_num

# Getter function to get the hourly pay rate.

def getHourlyRate(self):

return self.hourly\_rate

# Setter function to set the hourly pay rate.

def setHourlyRate(self, hourly\_rate):

self.hourly\_rate = hourly\_rate

'''

Deborah Barndt

4-8-19

EmployeeTest.py

hw10: Inheritance

This program will contain two classes: Employee and ProductionWorker. The Employee class

will keep data attributes for the following pieces of information: Employee name and

Employee number. The ProductionWorker will be a subclass of the Employee class. The

ProductionWorker class will keep data attributes for the following information: Shift

number (an integer, such as 1, 2, or 3) and Hourly pay rate.

The workday is divided into two shifts: day and night. The shift attribute will hold an

integer value representing the shift that the employee works. The day shift is shift 1 and

the night shift is shift 2. Write the appropriate accessor and mutator methods for each

class. Once you have written the classes, write a program that creates an object of the

ProductionWorker class and prompts the user to enter data for each of the object's data

attributes. Store the data in the object, then use the object's accessor methods to

retrieve it and display it on the screen.

Written by Deborah Barndt.

'''

from Employee import ProductionWorker

# Function to display the values of the data attributes.

def main():

while True:

print('Please enter the details of the employee.\n')

while True:

name = input('Enter employee name: ')

if not name:

print('\nInvalid input: Employee name cannot be empty.')

continue

else:

break

while True:

emp\_num = input('Enter employee number: ')

if not emp\_num:

print('\nInvalid input: Employee number cannot be empty.')

continue

try:

emp\_num = int(emp\_num)

except:

print('\nInvalid input: Please type a number.')

continue

else:

break

while True:

shift\_num = input('Enter shift number: ')

if not shift\_num:

print('\nInvalid input: Shift number cannot be empty.')

continue

try:

shift\_num = int(shift\_num)

except:

print('\nInvalid input: Please type a number.')

continue

else:

break

while True:

hourly\_rate = input('Enter hourly pay rate: ')

if not hourly\_rate:

print('\nInvalid input: Hourly pay rate cannot be empty.')

continue

try:

hourly\_rate = float(hourly\_rate)

except:

print('\nInvalid input: Please type a number.')

continue

else:

break

while True:

enterAgain = input('Would you like to enter another phone number? (y/n): ')

if not enterAgain:

print('\nInvalid input: Employee name cannot be empty.')

continue

else:

break

employee = ProductionWorker(name, emp\_num, shift\_num, hourly\_rate)

print('\n\n----------------------------------')

print(' Production Worker Details')

print('----------------------------------')

print('Employee Name: ', employee.getEmployeeName())

print('Employee Number: ', employee.getEmployeeNum())

if (employee.getShiftNum() == 1):

print('Shift Number: ', employee.getShiftNum(), 'Day shift')

elif (employee.getShiftNum() == 2):

print('Shift Number: ', employee.getShiftNum(), 'Night shift')

print('Hourly Pay Rate: $' + '%.2f' % employee.getHourlyRate())

print('----------------------------------\n\n')

if (enterAgain != 'y'):

print('Thank you for using the Production Worker Details program.')

break

# Call the main function to start the program.

main()

Output Result:







