Deborah Barndt

ITMD 513 Open Source Programming

Professor Dr. Sam

Hw11

4-19-19

'''

Deborah Barndt

4-15-19

Ship.py

hw11: Polymorphism

This program will contain three classes: Ship, CruiseShip, and CargoShip. The

program will demonstrate the classes with a list of Ships. The list elements

should be initialized with Ship, CruiseShip, and CargoShip objects. The program

should then step through the list, calling each object's print function, and

will use the Python issinstance() function to determine which object type each

element holds.

The Ship class has the following members: a member variable for the name of the

ship, a member variable for the year that the ship was built, a constructor and

appropriate accessors and mutators, and a print function that displays the

ship's name and the year it was built.

The CruiseShip class is derived from the Ship class, and will contain the

following members: a member variable for the maximum number of passengers,

a constructor and appropriate accessors and mutators, and a print function that

overrides the print function in the base class. The CruiseShip class's print

function should display only the ship's name and the maximum number of passengers.

The CargoShip class is derived from the Ship class, and will contain the

following members: a member variable for the cargo capacity in tonnage,

a constructor and appropriate accessors and mutators, and a print function that

overrides the print function in the base class. The CargoShip class's print

function should display only the ship's name and the ship's cargo capacity.

Written by Deborah Barndt.

'''

# Class Ship that contains the data for Ship name and the year it was built.

class Ship(object):

shipName = 'Titanic'

shipYear = 1912

# Function to create a constructor of class Ship.

def \_\_init\_\_(self, shipName, shipYear):

self.shipName = shipName

self.shipYear = shipYear

# Getter function to get ship name.

def getName(self, shipName):

return self.shipName

# Setter function to set the ship name.

def setName(self, shipName):

self.shipName = shipName

# Getter function to get the year the ship was built.

def getYear(self, shipYear):

return self.shipYear

# Setter function to set the year the ship was built.

def setYear(self, shipYear):

self.shipYear = shipYear

# Function to display the name of the ship and the year it was built.

def display(self):

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

print("The ship's name is: " + self.shipName)

print('The ship was built in: ' + self.shipYear)

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

# Class CruiseShip that contains the data for the maximum number of passengers.

class CruiseShip(Ship):

maxPass = 3000

# Function to create a constructor of class CruiseShip.

def \_\_init\_\_(self, maxPass):

self.maxPass = maxPass

# Getter function to get the maximum passengers.

def getMaxPass(self, maxPass):

return self.maxPass

# Setter function to set the maximum passengers.

def setMaxPass(self, maxPass):

self.maxPass = maxPass

# Function to display the name of the ship and the maximum passengers.

def display(self):

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

print("The ship's name is: " + self.shipName)

print('The maximum passengers are:', self.maxPass)

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

# Class CargoShip that contains the data for the cargo capacity in tonnage.

class CargoShip(Ship):

cargoCap = 20000

# Function to create a constructor of class CargoShip.

def \_\_init\_\_(self, cargoCap):

self.cargoCap = cargoCap

# Getter function to get the cargo capacity.

def getCargoCap(self, cargoCap):

return self.cargoCap

# Setter function to set the cargo capacity.

def setCargoCap(self, cargoCap):

self.cargoCap = cargoCap

# Function to display the name of the ship and the cargo capacity.

def display(self):

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

print("The ship's name is: " + self.shipName)

print('The cargo capacity of the ship is:', self.cargoCap, 'tons')

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

# Function to run the program for the ship, cruiseship, and cargoship classes.

def main():

# Function to display the ship list.

def showShips(ship):

if isinstance(ship, CargoShip):

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

print(' This is a cargo ship.')

ship.setName('Royal Caribbean')

ship.setCargoCap(100000)

ship.display()

elif isinstance(ship, CruiseShip):

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

print(' This is a cruise ship.')

ship.setName('Carnival')

ship.setYear('2011')

ship.display()

else:

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

print(' This is a ship.')

ship.display()

ships = ['ship', 'cruiseship', 'cargoship']

vessel = Ship('Titanic', '1912')

cruise = CruiseShip(4000)

cargo = CargoShip(50000)

ships = [vessel, cruise, cargo]

for ship in ships:

showShips(ship)

''' # Driver code for test 1.

ships = ['ship', 'cruiseship', 'cargoship']

ships[0] = Ship('Titanic', '1912')

ships[0].setName('Carnival')

ships[0].setYear('2011')

ships[1] = CruiseShip(3000)

ships[1].setName('Carnival')

ships[1].setMaxPass(4000)

ships[2] = CargoShip(20000)

ships[2].setName('Carnival')

ships[2].setCargoCap(50000)

# Call the display methods using the values in the list.

for x in ships:

x.display()

# Driver code for test 2.

ships = ['ship', 'cruiseship', 'cargoship']

ships[0] = Ship('Titanic', '1912')

ships[0].setName('Royal Caribbean')

ships[0].setYear('2001')

ships[1] = CruiseShip(3000)

ships[1].setName('Royal Caribbean')

ships[1].setMaxPass(4000)

ships[2] = CargoShip(20000)

ships[2].setName('Royal Caribbean')

ships[2].setCargoCap(100000)

# Call the display methods using the values in the list.

for x in ships:

x.display()

# Driver code for test 3.

ships = ['ship', 'cruiseship', 'cargoship']

ships[0] = Ship('Titanic', '1912')

ships[0].setName('Norwegian Getaway')

ships[0].setYear('2014')

ships[1] = CruiseShip(3000)

ships[1].setName('Norwegian Getaway')

ships[1].setMaxPass(3900)

ships[2] = CargoShip(20000)

ships[2].setName('Norwegian Getaway')

ships[2].setCargoCap(146000)

# Call the display methods using the values in the list.

for x in ships:

x.display()

# Driver code for test 4.

ships = ['ship', 'cruiseship', 'cargoship']

ships[0] = Ship('Titanic', '1912')

ships[0].setName('Majestic Princess')

ships[0].setYear('2017')

ships[1] = CruiseShip(3000)

ships[1].setName('Majestic Princess')

ships[1].setMaxPass(4300)

ships[2] = CargoShip(20000)

ships[2].setName('Majestic Princess')

ships[2].setCargoCap(145000)

# Call the display methods using the values in the list.

for x in ships:

x.display()

def showShips(vessel):

if isinstance(vessel, boats.Ship):

vessel.display()

else:

print('That is not a ship. Please try again.')

'''

# Call the main function to begin the program.

main()

Output Result:





