Instituto Tecnológico de Costa Rica

Escuela de Computación

Introducción a la Programación. I semestre 2018

Profesor: Jeff Schmidt Peralta

1. Escriba una función iterativa llamada `mayor_col(mat)` que recibe una matriz de tamaño `nxm`, devuelva un vector con los elementos mayores de cada una de las columnas de la matriz original.

```
>>> mayor_col([[0,1,2,4], [9,8,0,0], [5,0,3,2]])
[9, 8, 3, 4]
>>> mayor_col([[1, 2, 4], [3, 6, 8], [5, 10, 12]])
[5, 10, 12])
```

2. Escriba una función iterativa llamada `mayor(mat)` que recibe una matriz de tamaño `nxm`, devuelva un vector con los elementos mayores de cada una de las filas de la matriz original.

```
>>> mayor([[0,1,2,4], [9,8,0,0], [5,0,3,2]])
[4, 9, 5]
>>> mayor([[1, 2, 4], [3, 6, 8], [5, 10, 12]])
[4, 8, 12])
```

3. Herencia. Se desea modelar un problema para manejar objetos con un nivel de jerarquía. Se va a definir una clase padre llamada vehiculo que va a contener: placa (string), marca, cantidad de ruedas, kilometraje y consumo por km. Esta clase padre va a tener varios métodos: mostrar, hacerViaje(kms) que actualiza el kilometraje del vehículo.

   A su vez se van a definir 2 subclases, la primera llamada auto, que además de los atributos definidos en la clase vehiculo, tiene marca, modelo, combustible. Esta clase va a tener un método mostrar. La segunda subclase se llama moto y va a tener además de los atributos heredados los siguientes: estilo, cilindraje y va tener un método mostrar.

   i. Defina las clases Vehiculo, Auto y Moto.
   ii. Defina una lista de instancias de ambos vehículos. Muestre los datos de los vehículos que tengan más de 10,000 kms.

4. Se desea construir un objeto tipo uber, los atributos son: el número de placa, marca, año, estado del vehículo (libre, ocupado, reparación), tipo (económico, premium), cantidad de viajes realizados, monto de viaje (el actual) y acumulado de monto por viajes. Los métodos a implementar son: `mostrar()`, `reset()` que pone en cero la cantidad de viajes realizados, `enviar_ reparación()` que actualiza el estado del camión a reparación, `recibir_ reparación()` que actualiza el estado del camión a libre, `viaje(kilometros)` que aumenta en 1 la cantidad de viajes realizados y calcula el monto a pagar por el viaje, de acuerdo a la cantidad de kilómetros de viaje. Considere que los viajes económicos se pagan a 300 por kilómetro y los Premium a 500 por kilómetro. Además deben existir los métodos `get_tipo()`, `get_estado()` y `get_viajes()` que retornan los datos correspondientes.

   a. Defina una clase `Uber` para el manejo de este objeto.

   b. Luego de tener definida la clase, se asume que se tiene la lista de instancias y que se quiere construir una función llamada `solicita(lista, tipo)`, que recibe esa lista de instancias para saber cuál o cuáles vehículos podrán transportar un cliente según un tipo de vehículo.

5. Asuma la siguiente definición parcial de una clase Lista, que va a representar una lista simple:

```
class Nodo:
      def __init__(self, valor = None):
            self.next = None
            self.valor = valor
class Lista:
      def __init__(self):
            self.head = None
            self.largo = 0
```

Escriba los siguientes métodos. En caso de requerir algún otro método adicional en alguna de las dos clases, debe escribirlo.

`inse(valor1, valor2):` inserta `valor1` en un nodo antes de la primera aparición de `valor2`

`prom():` saca promedio de la lista.

`dela(valor):` elimina todas las apariciones del nodo en la lista.

6. Herencia. Se desea modelar un problema para manejar objetos con un nivel de jerarquía. Se va a definir una clase padre llamada vehiculo que va a contener: placa (string), marca, cantidad de ruedas, kilometraje y consumo por km. Esta clase padre va a tener varios métodos: mostrar, hacerViaje(kms) que actualiza el kilometraje del vehículo.

   A su vez se van a definir 2 subclases, la primera llamada auto, que además de los atributos definidos en la clase vehiculo, tiene marca, modelo, combustible. Esta clase va a tener un método mostrar. La segunda subclase se llama moto y va a tener además de los atributos heredados los siguientes: estilo, cilindraje y va tener un método mostrar.

   i. Defina las clases Vehiculo, Auto y Moto.
   ii. Defina una lista de instancias de ambos vehículos. Muestre los datos de los vehículos que tengan más de 10,000 kms.

7. Se desea construir un objeto tipo uber, los atributos son: el número de placa, marca, año, estado del vehículo (libre, ocupado, reparación), tipo (económico, premium), cantidad de viajes realizados, monto de viaje (el actual) y acumulado de monto por viajes. Los métodos a implementar son: `mostrar()`, `reset()` que pone en cero la cantidad de viajes realizados, `enviar_ reparación()` que actualiza el estado del camión a reparación, `recibir_ reparación()` que actualiza el estado del camión a libre, `viaje(kilometros)` que aumenta en 1 la cantidad de viajes realizados y calcula el monto a pagar por el viaje, de acuerdo a la cantidad de kilómetros de viaje. Considere que los viajes económicos se pagan a 300 por kilómetro y los Premium a 500 por kilómetro. Además deben existir los métodos `get_tipo()`, `get_estado()` y `get_viajes()` que retornan los datos correspondientes.

   a. Defina una clase `Uber` para el manejo de este objeto.

   b. Luego de tener definida la clase, se asume que se tiene la lista de instancias y que se quiere construir una función llamada `solicita(lista, tipo)`, que recibe esa lista de instancias para saber cuál o cuáles vehículos podrán transportar un cliente según un tipo de vehículo.

8. Un negocio vende CDs y DVDs. Cada uno de estos artículos tiene un tipo (CD o DVD), título, precio, duración en minutos, autor y ventas en unidades (inicia en cero cuando se crea una instancia). Se desea construir un objeto Articulo con los datos indicados. Los métodos a implementar son: `mostrar`, que muestra todos los datos de un artículo, `venta` que recibe una cantidad y actualiza (suma) las ventas del artículo, `devolucion` que recibe una cantidad y actualiza (resta) las ventas del artículo.
    a. Defina una clase `Articulo` para el manejo de estos objetos.
    b. Luego de tener definida la clase, se asume que se tiene una lista de instancias y que se quiere construir una función llamada `top`, para saber cuál es el DVD que registra más ventas y cuál es el DVD que registra más devoluciones. En este caso se recibiría como argumento de la función la lista de instancias.