Situación problemática

En una bodega se desea guardar la información de cada uno de los 20 camiones que utilizan para trasladar la producción durante la cosecha que transcurre en un lapso de 45 días corridos.

a- Defina la clase camión que posea como atributos: un identificador (un valor numérico de 1 a 20), nombre del conductor, patente del camión, marca del camión y tara (peso del camión vacío).

b- Defina una clase cosecha que contenga como atributo una lista bidimensional que permita registrar la cantidad de kilos por día que cada camión descarga en la bodega. El ingreso es a través de un archivo de texto con los datos en el siguiente orden: identificador de camión, día, y el peso del camión cargado (el peso que informa la báscula de la bodega), para obtener los kilogramos de uva descargados se calcula la diferencia entre el peso del camión cargado y la tara. Un camión puede realizar varios viajes desde la finca a la bodega.

c- Implemente un programa que:

1- Lea los datos desde un archivo separado por comas y genere las instancias correspondientes de la clase “Camión”, almacenándolas en una lista.

2- Leer los datos de la cosecha desde un archivo separado por comas.

3- Presente un menú de opciones permita realizar las siguientes tareas:

3-1- Dado el número de identificador de un camión mostrar, la cantidad total de kilos descargados.

3-2- Dado un número correspondiente a un día mostrar un listado con el siguiente formato.

PATENTE	CONDUCTOR	CANTIDAD DE KILOS
********	**********	**********
*********	**********	**********

Solución

Clase Camión

```
class Camion:
    __ID = 0
    __Conductor = ''
    __Patente = ''
    __Marca = ''
    __Tara = 0
    def __init__(self, ID, Conductor, Patente, Marca, Tara = 0):
        self.__ID = ID
        self.__Conductor = Conductor
        self.__Patente = Patente
        self.__Marca = Marca
        self.__Tara = Tara
    def __str__(self):
        return "%d %s %s %s %d" %
(self.__ID,self.__Conductor,self.__Patente,self.__Marca,self.__Tara)
    def getPatente(self):
        return self.__Patente
    def getConductor(self):
        return self.__Conductor
    def getTara(self):
        return self.__Tara
```

Clase Cosecha

```
from Camion import Camion
from ManejaCAM import ManejaCamion
import csv

class Cosecha:
    def __init__(self,C,D):
        self.__listaCosecha = [[0.0 for _ in range(D)] for _ in
range(C)]
    def __str__(self):
        print(self.__listaCosecha)

    def agregarDescarga(self,cam,dia,pesoNeto):
        self.__listaCosecha[cam-1][dia-1] += pesoNeto

    def __str__(self):
        s = ""
        for Cosecha in self.__listaCosecha:
            s += str(Cosecha) + '\n'
        return s

    def getDescargasCamion(self,camion):
        return self.__listaCosecha[camion-1]

    def getListaCosecha(self):
        return self.__listaCosecha

    def obtenerKGDescargados(self, camion):
        suma=0.0
        filaCamion = self.getDescargasCamion(camion)
        for i in range (len(filaCamion)):
            suma += filaCamion[i]
        return suma

    def informeCosechaPorDia(self, dia, mCa):
        print('PATENTE    CONDUCTOR     CANTIDAD DE KILOS')
        for i in range(mCa.getCantCamiones()):
            if self.__listaCosecha[i][dia-1] != 0:
                unCamion=mCa.getCamion(i+1)
                print('{}     {}
{}'.format(unCamion.getPatente(),unCamion.getConductor(),self.__listaC
osecha[i][dia-1]))

    def testCosecha(self,mCa):
        archivo = open('cosechaEJ3.csv')
        reader = csv.reader (archivo, delimiter = ',')
        for fila in reader:
            cam = int(fila[0])
            dia = int(fila[1])
            peso = float(fila[2])
            pesoNeto = peso - mCa.getCamion(cam).getTara()
            self.agregarDescarga(cam,dia,pesoNeto)
        archivo.close()
```

Clase ManejadorCamion

```
from Camion import Camion
import csv

class ManejaCamion:
    def __init__(self):
        self.__listaCamiones = []
```

```
    def agregarCamion (self, unCamion):
        self.__listaCamiones.append(unCamion)

    def getCamion(self,camion):
        ca = camion - 1
        return self.__listaCamiones[ca]

    def getCantCamiones(self):
        return len(self.__listaCamiones)


    def testCamiones(self):
        archivo = open('camionesEJ3.csv')
        reader = csv.reader (archivo, delimiter = ',')
        for fila in reader:
            ID = int(fila[0])
            Conductor = fila[1]
            Patente = fila[2]
            Marca = fila[3]
            Tara = int(fila[4])
            unCamion = Camion(ID, Conductor, Patente, Marca, Tara)
            self.agregarCamion(unCamion)
        archivo.close()

    def __str__(self):
        s = ""
        for Camion in self.__listaCamiones:
            s += str(Camion) + '\n'
        return s
```

Programa Principal

```
from Camion import Camion
from ManejaCAM import ManejaCamion
from Cosecha import Cosecha


def opcion0(Co,mCa):
    print('Adios')

def opcion1(Co,mCa):
    print ('Ingrese el número de Camion a consultar: ')
    CAM = int(input())
    sumaKG=Co.obtenerKGDescargados(CAM)
    print('El total de KG descargados por el camion consultado es
{}'.format(sumaKG))

def opcion2(Co,mCa):
    print ('Ingrese el día para generar informe: ')
    DIA = int(input())
    Co.informeCosechaPorDia(DIA, mCa)

switcher = {
    0: opcion0,
    1: opcion1,
    2: opcion2
}

def switch(argument,mv,indice):
    func = switcher.get(argument, lambda: print("Opción incorrecta"))
    func(mv,indice)
```

```
if __name__ == '__main__':
    print('Ingrese la Cantidad de Camiones: ')
    C=int(input())
    print('Ingrese la Cantidad de Dias de la cosecha: ')
    D=int(input())
    mCa = ManejaCamion()
    mCa.testCamiones()
    print(mCa)
    Co = Cosecha(C,D)
    Co.testCosecha(mCa)
    print(Co)

    bandera = False
    while not bandera:
        print("\nMENU DE OPCIONES")
        print("0 Salir")
        print("1 ITEM 3.1: Mostrar la Cantidad de kg descargados por
un camión")
        print("2 ITEM 3.2: Mostrar Listado de descargar para un día")
        opcion = int(input ('Ingrese una opcion:'))
        switch(opcion, Co,mCa)
        bandera = int(opcion)== 0
```