

Examen

Final de Septiembre

12 de Septiembre de 2014

▷ 1. Organización de una liga

(4 puntos) De todos es conocida la prestigiosa liga de la Federación Española de Jürgol. Este año, para poder sacar más tajada fomentar la práctica deportiva, van a montar un chiringuito de apuestas construir un casino benéfico, basado en los resultados de la competición. Para poder afinar bien las pujas quieren hacer un simulador de la liga.

El programa de simulación debe generar los emparejamientos al comienzo, y luego simular cada jornada del campeonato. Por cada jornada se debe almacenar el resultado de cada partido. Finalmente se mostrará un resumen en el que se mostrará la clasificación jornada a jornada.

Para determinar el calendario del torneo, se realiza mediante un simple algoritmo de selección. A cada uno de los competidores se le asigna un número (de forma aleatoria) y se van rotando los números, dejando uno fijo. Por ejemplo, en un torneo con 14 competidores:

Jornada 1

1	2	3	4	5	6	7
14	13	12	11	10	9	8

Jornada 2

1	14	2	3	4	5	6
13	12	11	10	9	8	7

Jornada 3

1	13	14	2	3	4	5
12	11	10	9	8	7	6

.....

Jornada 13

1	3	4	5	6	7	8
2	14	13	12	11	10	9

Para simular un partido se usará la habilidad de cada equipo. La habilidad no es más un número real que indica lo bueno que es cada equipo. Este número será usado para simular una partida de la siguiente forma: supongamos que el equipo a tiene una habilidad h_a y el equipo b una habilidad h_b ; entonces se genera un número aleatorio x distribuido de forma uniforme entre 0 y $h_a + h_b$; entonces

- Si $x < h_a - \frac{\min(h_a, h_b)}{3}$ gana a .
- Si $h_a - \frac{\min(h_a, h_b)}{3} \leq x \leq h_a + \frac{\min(h_a, h_b)}{3}$ empatan.
- Si $x > h_a + \frac{\min(h_a, h_b)}{3}$ gana b .

- (2 puntos) Haz un programa (función, procedimiento o método) que, con el algoritmo dado, genere un calendario para una liga de n equipos donde n es par.

- **(2 puntos)** Haz un programa (función, procedimiento o método) que dado un calendario y una jornada (un número entre 1 y $n - 1$), simule dicha jornada y devuelva los resultados de los partidos.

▷ **2. Monstruos S.A.**
(6 puntos)

Monstruos S.A. es la compañía eléctrica más importante de Monstruópolis. Su método de producción se basa en emparejar cada niño humano con el monstruo asustador más adecuado, siguiendo unos criterios estrictos de compatibilidad. De este modo, se obtienen gritos de la mejor calidad, que una vez refinados y tratados se convierten en energía limpia y fiable. Sin embargo, últimamente los asustadores han podido comprobar que realizar su trabajo les resulta muy complicado, ya que cada vez es más difícil asustar a los humanos. Como consecuencia de la escasez de gritos, la energía eléctrica se ha convertido en un producto escaso en Monstruópolis, lo que ha obligado a las autoridades a controlar su consumo, sobre todo en el parque automovilístico de la ciudad. Por este motivo, y con el propósito de limitar el gasto por habitante, se ha decidido elaborar una base de datos que ha de almacenar la información siguiente.

1. Solo algunos modelos de ciertas marcas de automóviles pueden transitar por las carreteras de Monstruópolis. Por esta razón, se guarda en una estructura de datos toda la información de dichas **especificaciones**, que incluye la **marca**, el **modelo** y su **consumo medio** por cada 100 km (medido en kWh).
2. Los **vehículos** del parque automovilístico se identifican unívocamente mediante su número de **matrícula**. Durante el proceso de registro, también se pide al propietario que aporte los datos correspondientes al **color** y la **especificación** (*marca*, modelo y consumo medio). Además, todo vehículo se matricula a nombre de un único ciudadano mayor de edad, aunque es posible que éste tenga en propiedad varios automóviles.
3. La información sobre los **propietarios** se compone de su **NIF**, el **nombre**, **primer apellido** y **edad**, la fecha de **caducidad** de su carnet de conducir y una dirección válida de **e-mail**. Por ejemplo, Mike Wazowski es un propietario de 25 años. Su NIF es 69696969W, su carnet de conducir caduca el 23-03-2015 y su e-mail es ojitoSalton@monstruos.sa. Este monstruo ha matriculado un Nissan Moco verde cuyo consumo medio es de 18 kWh/100 km y un Lamborghini Reventón rojo algo más eficiente (consumo medio de 14,4 kWh/100 km).

AmpliacionMultasAcabas de sustituir al programador que estaba implementando el sistema de gestión del parque automovilístico de Monstruopolis.

Su código inicial es el siguiente:

```
class especificacion(object):
    def __init__(self,marca,modelo,consumo):
        '''
        mairca:string
        modelo:string
        consumo:float
        '''
        self.marca=marca
        self.modelo=modelo
        self.consumo=consumo
    def __eq__(self,otra):
        '''
```

```

        comprueba la igualdad de dos especificaciones
        '''
        return self.marca==otra.marca and self.modelo==otra.modelo and self.consumo==

class vehiculo(object):
    def __init__(self,matricula,color,especif,prop,lista_espec):
        '''
        matricula:str
        color:string
        especific:especificacion
        prop:propietario
        lista_espec: lista de especificaciones
        '''
        if especific not in lista_espec:
            raise Exception('El coche con especificacion: '+str(especific)+' no puede c
        else:
            self.matricula=matricula
            self.color=color
            self.especific=especific
            self.prop=prop

class permiso_conducir(object):
    def __init__(self,fecha_cad,puntos):
        self.fecha=fecha_cad
        self.puntos=puntos

class propietario(object):
    def __init__(self,nif,nombre,apellido,permiso,edad,email):
        self.nif=nif
        self.nombre=nombre
        self.apellido=apellido
        self.permiso=permiso
        self.edad=edad
        self.email=email

class parque_automoviles(object):
    def __init__(self):
        self.listaPopietarios=[]
        self.listaVehiculos=[]
    def registrar_vehiculo(self,v):
        if v.propietario not in self.listaPopietarios:
            self.listaPopietarios.append(v.prop)
        if v not in listaVehiculos:
            self.listaVehiculos.append(v)

```

Aparte de la gestión de los parámetros erróneos en las definiciones de los métodos generando las correspondientes excepciones, te tienes que poner con urgencia a implementar otra funcionalidad al sistema. La policía municipal de Monstruópolis quiere gestionar las **multas**. La información de cada multa consta de la matrícula del vehículo con el que se ha cometido la infracción, el importe (en euros) de la multa y el número de puntos a retirar del permiso de circulación del infractor.

- Implementa la clase **multa** para poder ampliar el sistema. Cada multa tiene información sobre el vehículo infractor, el importe en euros y el número de puntos de sanción a descontar en el permiso de circulación del infractor.

Implementa el método `procesa_multas` dentro de la clase `parque_automoviles` que, dada una lista de multas:

1. actualice los puntos del permiso de conducir de los propietarios de los vehículos con los que se ha cometido una infracción. Si un propietario pierde todos los puntos de su permiso de conducir todos sus vehículos serán retenidos, debes modificar la clase `parqueautomoviles` para poder guardar la información de los vehículos retenidos
2. calcule y devuelva la información del importe que cada propietario adeuda al Ayuntamiento de Monstuópolis.

Tiempo para resolver el examen: 3 horas