

Examen de Programación Septiembre 2020 (Grados en Física y Matemáticas)

Primera parte (1.25 puntos por cuestión, 50% nota del examen)

- 1) Se desea escribir una función que retorne una tupla con los siguientes datos relativos al string que se pasa como parámetro: a) El número de espacios en blanco. b) El número de signos de puntuación que sean '.', ',', ' ' o ';'. Por ejemplo, si el string es "Una prueba, fácil.", se deberá retornar (2,2).
- 2) La siguiente tabla muestra las tarifas de una compañía de telefonía móvil:

Nombre tarifa	Límite de datos mensuales	contrato	pre-pago
Básica	3 GB	11,20 €	12,00 €
Media	20 GB	20,00 €	24,00 €
Avanzada	40 GB	25,60 €	30,00 €
Infinita	Ilimitado	32,00 €	No disponible

Escribir una función a la que se le pasen como parámetros el número de gigabytes mensuales deseados (número real), y un booleano indicando si se desea una tarifa de contrato o no. La función retornará una tupla con el nombre de la tarifa aplicable y el precio mensual. La tarifa aplicable es la más barata que cubra los gigabytes deseados, por ejemplo si se desean 15GB la tarifa sería la de 20GB. Si no hubiese ninguna tarifa aplicable (caso de pre-pago con más de 40GB) se retornará como nombre de tarifa "No disponible" y precio mensual `math.nan`. Se valorará la eficiencia y tamaño compacto de la solución.

- 3) Escribir una función que calcule y retorne el valor de la función $\cos(x)$ dado por la siguiente expansión en serie:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} + \dots$$

La función recibe como parámetros el valor de x y el número de términos del desarrollo en serie, n , numerándose los términos desde 0 hasta $n-1$. Por ejemplo, la fórmula de arriba correspondería al caso $n=5$.

Por eficiencia se debe evitar usar el operador `**` o las función `pow()` o `factorial()`. Para ello, el numerador del término i -ésimo (para i desde 1 hasta $n-1$) del desarrollo en serie se debe obtener del numerador anterior multiplicándolo por $-x^2$ y el denominador será el anterior por $2i \cdot (2i - 1)$. Obviamente, el numerador y denominador del primer término (el correspondiente a $i=0$) son ambos iguales a 1.

- 4) En un computador con sistema operativo Linux disponemos de un directorio llamado `data` situado en el directorio del usuario y que contiene tres tipos de ficheros: de texto, imágenes

png y ficheros de log (registros de actividad). Por ejemplo, el contenido podrían ser estos ficheros: (list1.txt, list2.txt, list3.txt, img1.png, img2.png, img.log). Indica el contenido del script que crearíamos para realizar los siguientes pasos, suponiendo que partimos del directorio de trabajo igual al del usuario

- crear en el directorio del usuario un directorio nuevo llamado backup
- mover al directorio backup los ficheros de data acabados en .txt
- copiar en el directorio backup los ficheros de data acabados en .png
- crear en el directorio backup un nuevo directorio llamado logs
- copiar en logs los archivos de data acabados en .log
- borrar el directorio data con todos sus contenidos

Fases a seguir:

- En primer lugar hacer un esquema de la distribución final de directorios y ficheros usando el ejemplo indicado.
- En segundo lugar escribir el *script*.

Examen de Programación Septiembre 2020 (Grados en Física y Matemáticas)

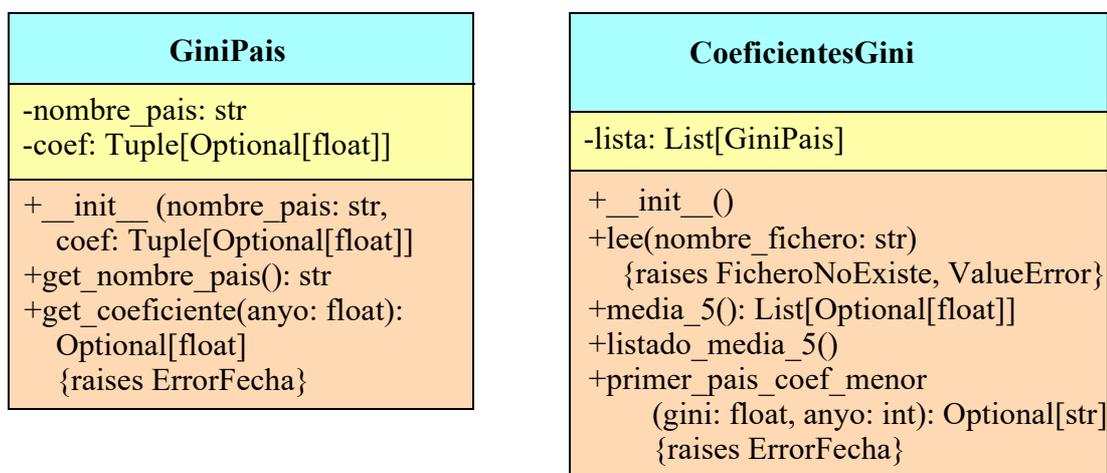
Segunda parte (5 puntos, 50% nota del examen)

Se desea hacer parte del software de análisis de los datos almacenados en un fichero que contiene coeficientes de Gini de diversos países (https://es.wikipedia.org/wiki/Coeficiente_de_Gini). Este coeficiente es una medida de la desigualdad de los ingresos de las personas de un país. Toma valores entre 0 y 100, siendo los valores bajos indicativos de poca desigualdad. Por el contrario, los valores altos reflejan una gran desigualdad.

Se dispone del módulo `gini.py` que contiene un par de constantes que reflejan el intervalo de años para los que hay datos registrados, así como las clases `GiniPais` y `CoeficientesGini` y las excepciones `ErrorFecha` y `FicheroNoExiste`. Las constantes son:

- `PRIMER_ANYO` (int): el primer año del intervalo para el que hay datos registrados (incluido)
- `ULTIMO_ANYO` (int): el último año del intervalo para el que hay datos registrados (incluido).

La clase `GiniPais`, ya realizada, contiene el nombre de un país y los coeficientes Gini de ese país entre los años 1967 y 2018, ambos incluidos, así como métodos para obtener estos datos. Responde al diagrama de clases que se muestra



Los atributos de la clase se indican a continuación:

- `nombre_pais`: el nombre del país
- `coef`: Tupla de números reales correspondientes a los coeficientes de ese país entre `PRIMER_ANYO` y `ULTIMO_ANYO`, ambos incluidos. Algunos coeficientes no se conocen y en ese caso en lugar de un número real se almacena el dato `None`. Esto se refleja por medio de la anotación de tipo `Optional[float]`, que indica que el dato es o bien un `float` o bien el valor `None`.

Los métodos de la clase `GiniPais` son:

- `get_nombre_pais()`: retorna el nombre del país

- `get_coeficiente()`: Retorna el coeficiente Gini del año indicado (que puede ser `None`, si no se conoce). Si el año no está en el intervalo entre `PRIMER_ANYO` y `ULTIMO_ANYO` se lanza la excepción `ErrorFecha`

Se pide hacer en el mismo módulo `gini.py` parte de la clase `CoeficientesGini` que responde al diagrama de clases que se muestra arriba. La clase almacena en un atributo una lista de objetos de la clase `GiniPais`. También dispone de diversos métodos para analizar los coeficientes:

- *constructor*: crea la lista vacía guardándola en el atributo `lista`
- `lee()`: (Método ya hecho) lee los datos de los coeficientes de Gini del fichero cuyo nombre se halla en el atributo `nombre_fichero` y los añade al atributo `lista`. Si el fichero no existe, lanza la excepción `FicheroNoExiste`. Si el fichero tiene errores lanza `ValueError`, pero supondremos que el fichero es correcto e ignoraremos esta excepción.
- `media_5()`: Calcula la media de los últimos cinco años registrados (desde `ULTIMO_ANYO-4` hasta `ULTIMO_ANYO`, ambos incluidos) para cada país de la lista original y retorna una lista con esas medias, un elemento por cada país. Este elemento es `None` si ese país no tiene ningún coeficiente Gini en los últimos 5 años registrados (es decir, si todos valen `None`). En otro caso es la media de los cinco últimos años registrados, ignorando los que valen `None`. Es decir, si hay `N` años con coeficientes válidos (que no sean `None`), es la media de esos `N` coeficientes. Por ejemplo, si los cinco coeficientes son `[5, None, None, 3, 4]` la media será $(5+3+4)/3 = 4$.
- `listado_media_5()`: Hace un listado en pantalla de la media de los últimos 5 años de los coeficientes Gini de todos los países de la lista, de acuerdo a lo obtenido con el método `media_5()`. Para hacer el listado:
 - En primer lugar se muestra un encabezamiento de dos líneas explicando lo que contiene el listado y las dos columnas que vienen a continuación.
 - A continuación se muestra en cada línea un coeficiente. Si es un número se muestra con anchura fija y un decimal; si vale `None` se muestra el texto `None` (con la misma anchura que los números). Luego, en la misma línea, se muestra el nombre del país.
 - Al finalizar el listado se muestran dos líneas en blanco para separarlo de lo siguiente que se muestre en pantalla. Ejemplo de las primeras líneas del listado:

```
Listado del coeficiente Gini medio de los últimos 5 años
Media País
33.6 Albania
None Algeria
51.3 Angola
```

- `primer_pais_coef_menor()`: Retorna el nombre del primer país de la lista cuyo coeficiente Gini es inferior al parámetro `gini` en el año indicado por `anyo`. Se ignoran los valores `None`. Si ningún país cumple el criterio se retorna `None`. Lanza `ErrorFecha` si el parámetro `anyo` no está entre `PRIMER_ANYO` y `ULTIMO_ANYO`

Finalmente, se pide hacer un programa principal en el mismo módulo, que haga lo siguiente:

- a. Crea un objeto de la clase `CoeficientesGini`
- b. Invoca al método `lee()` con el nombre de fichero `inequality_index_gini.csv`, para rellenar la lista
- c. Muestra en pantalla un listado de los coeficientes medios de los últimos 5 años

d. Muestra en pantalla el primer país con coeficiente menor que 25 en 2017

e. Pone en pantalla "Fin de la prueba"

Tratamiento de errores a realizar en el programa principal:

- Si en el paso b) se lanzase `FicheroNoExiste` se abandonan los restantes pasos y se escribe el mensaje "El programa no se puede ejecutar"
- Si en el paso d) se lanzase `ErrorFecha` se pone un mensaje de error y luego se continúa normalmente con el paso e)

Valoración (sobre 5):

- encabezamiento de la clase y constructor: 0.5 puntos
- basta entregar 2 de estos 3 métodos, o en caso de entregar los 3 se valoran los 2 mejores:
 - método `media_5()`: 1.5 puntos
 - método `listado_media_5()`: 1.5 puntos
 - método `primer_pais_coef_menor()`: 1.5 puntos
- programa principal: 1.5 puntos