



Actividad 02

Programación Funcional

Introducción

El malvado Dr.Herny ha derrocado a nuestra Reina Barrios arrebatándole los [6 lenguajes del infinito](#) (otorgados por un misterioso [LanguageKeeper](#)), con los cuales planea destruir ~~sus promedios~~ la mitad de la humanidad. Como sabes que su poder será implacable (y quieres sobrevivir), no te queda más que ayudarle entregándole estadísticas sobre su base de datos de la humanidad, para así tomar una buena decisión al momento de desintegrarnos.

El objetivo de esta actividad es realizar un sistema de consultas, utilizando **programación funcional**, que permita obtener información sobre las personas, ciudades y países de nuestro planeta. Para realizar lo anterior, debe guardar la información de los archivos en una estructura de datos adecuada y completar las funciones entregadas en el archivo `main.py`. **Está prohibido el uso de `for` y `while` excepto para generadores, funciones generadoras, listas y generadores por comprensión**. Para imprimir resultados en consola deberán utilizar la función `foreach`, que estará definida en el archivo que se les entregó.

Archivos

Contienen la información del nuevo universo. A continuación se detalla el contenido de los archivos:

- `Ciudades.txt` Contiene la información de las ciudades. Cada fila es de la forma:
`sigla_país,nombre`
- `Países.txt` Contiene la información de los países. Cada fila es de la forma:
`sigla,nombre`
- `Personas.txt` Contiene la información de las personas. Cada fila es de la forma:
`nombre,apellido,edad,sexo,ciudad_de_residencia,área_de_trabajo,sueldo`
donde `ciudad_de_residencia` corresponde al nombre de la ciudad a la cual se hace referencia.

Funciones

Lectura

Para leer los archivos que les entregamos, debes hacer uso eficiente de la memoria. Es por esto que debes crear las siguientes funciones generadoras¹. Para facilitarles la estructura que deben retornar en las siguientes funciones, en el archivo `main.py` encontrarán 3 namedtuples que contienen la estructura correspondiente a `Ciudad`, `Pais` y `Persona`.

- `def leer_ciudades(ruta_archivo_ciudades)`
 - **Recibe:** la ruta al archivo `Ciudades.txt`.
 - **Retorna:** un generador de ciudades.
- `def leer_paises(ruta_archivo_paises)`
 - **Recibe:** la ruta al archivo `Paises.txt`.
 - **Retorna:** un generador de países.
- `def leer_personas(ruta_archivo_persona)`
 - **Recibe:** la ruta al archivo `Personas.txt`.
 - **Retorna:** un generador de personas.

Consultas

Llegó el turno de trabajar con consultas sobre los datos leídos anteriormente. En estas funciones deberás trabajar con iterables como parámetros y retornar un generador.

- `def sigla_de_pais(nombre_pais, paises)`
 - **Recibe:** nombre de un país y un iterable con países.
 - **Retorna:** Un `str` con la sigla de dicho país.
- `def ciudades_por_pais(nombre_pais, paises, ciudades)`
 - **Recibe:** nombre de un país, un iterable con países, y un iterable con ciudades.
 - **Retorna:** un generador de las ciudades pertenecientes al país entregado.
- `def personas_por_pais(nombre_pais, paises, ciudades, personas)`
 - **Recibe:** nombre de un país, un iterable con ciudades, y un iterable con personas.
 - **Retorna:** un generador con las personas pertenecientes al país entregado.
- `def sueldo_promedio(personas)`
 - **Recibe:** un iterable con personas.
 - **Retorna:** Un valor `int/float`. No es necesario considerar el caso en que el iterable no contiene elementos.

¹Las funciones generadoras retornan valores usando la palabra reservada `yield`

- `def cant_personas_por_area_de_trabajo(personas)`
 - **Recibe:** un iterable con personas.
 - **Retorna:** un diccionario de la forma `{area_de_trabajo: int}`

Notas

- La librería `statistics` puede resultarles útil.
- Asume que los nombres de ciudades son únicos, es decir el nombre de la ciudad no se repite en ningún otro país del mundo.
- No modifique los archivos `.txt` entregados.

Requerimientos

- `leer_ciudades`
- `leer_paises`
- `leer_personas`
- `ciudad_por_pais`
- `personas_por_pais`
- `sueldo_promedio`
- `cant_personas_por_area_de_trabajo`

Entrega

- **Lugar:** En su repositorio privado de GitHub, en la **carpeta** `Actividades/AC02/`
- **Hora del último *push*:** 16:00