



Actividad 03

Programación funcional y decoradores

Introducción

Luego de salvar al mundo de los malvados planes de Dr.H, decides descansar de tu ardua tarea. Mientras descansas un amigo tuyo que trabaja en Chivelisión (sic) te pide ayuda para un nuevo programa llamado “Pasaprogra”. Tu amigo además te recomienda leer por lo menos hasta la sección “Lo que tienes que hacer” de este enunciado antes de empezar a trabajar.

Pasaprogra

Es un juego basado en el famoso programa de televisión Pasapalabra. A diferencia de este, Pasaprogra es individual y contiene tan solo dos juegos:

1. **Ahorcado:** En este juego se le entrega al jugador una oración con una palabra inconclusa. Este debe intentar adivinar dicha palabra.

Ejemplo: La tarea de progra estaba muy f _ c _ _ .

El jugador debe adivinar la palabra ‘facil’. Notar que las palabras que deben ser adivinadas no tienen tilde, aunque tengan que llevarlo por las normas del español.

2. **Rosca:** En este juego se le entrega una definición al jugador. Luego este debe intentar adivinar qué palabra es la que se está describiendo.

Ejemplo: Mueble formado por un tablero horizontal, sostenido por uno o varios pies, con la altura conveniente para poder realizar alguna actividad sobre ella o dejar cosas encima.

El jugador debería responder ‘mesa’.

En ambos juegos el jugador debe ingresar su respuesta y luego se revisa si la respuesta es correcta. Si respondió correctamente se le agregará 1 punto, y en caso de estar equivocado se le quitará 1 punto. Al igual que en el conocido Pasapalabra, si el jugador ingresa como respuesta “pasaprogra” pasará a la siguiente pregunta sin que se le otorgue o descuenta puntaje.

Archivos y encriptación

Cada juego tiene un archivo con sus respectivas preguntas. A continuación se muestran los formatos de los archivos:

- **Ahorcado:** Para este juego encontrarás un archivo `ahorcado_encriptada.txt`. Cada línea contiene dos elementos separados por una coma (","): la palabra correcta y oración. En el caso de la oración, la palabra correcta aparece reemplazada por "palabra".

Ejemplo:

`epoca,el 18 es la mejor palabra del año.`

- **Rosca:** Para este juego encontrarás un archivo `rosca_encriptada.txt`. Cada línea contiene dos elementos separados por una coma (","): la palabra correcta, y la definición de esa palabra.

Ejemplo:

`pipeño,vino tradicional chileno de cepas no nobles ...`

Dado que hay participantes mal intencionados, Chivelisión encriptó los archivos. La encriptación consiste en que cada palabra se modifica bajo la clave `arquetipos`. Esta clave consiste en intercambiar las letras de la palabra `arquetipos` por los primeros diez números, incluido el cero, tal como se muestra a continuación:

A	R	Q	U	E	T	I	P	O	S
0	1	2	3	4	5	6	7	8	9

Cuadro 1: Clave arquetipos

Por ejemplo, al encriptar `'zapato'` utilizando el patrón `arquetipos` obtenemos la palabra `'z07058'`. También vale notar que si se encripta `'hernan444'`, los números también se cambian por las letras a las que están asociadas, por lo que se obtendría `'h41n0neee'`. Este hecho genera una propiedad interesante: el proceso de encriptación y desencriptación es exactamente el mismo.

Código

Lo que ya está

En el archivo `main.py` se encuentra el juego implementado. Lo que debes hacer es usar `@decoradores` para completar ciertas funcionalidades del programa. El programa está constituido por las siguientes funciones:

- `def leer_archivo(ruta_archivo):` Esta **función generadora** recibe una ruta o *path* a un archivo y genera los elementos de este.
- `def juego_ahorcado(ruta_archivo_ahorcado):` Esta **función generadora** recibe el *path* al archivo con las preguntas del ahorcado. Genera dos elementos cada vez: la palabra correcta y su oración asociada.
- `def juego_rosca(ruta_archivo_rosca):` Esta **función generadora** recibe el *path* al archivo con las preguntas del juego rosca. Genera dos elementos cada vez: la palabra correcta y su definición asociada.

- `def jugar(respuesta, palabra_correcta)`: Esta función recibe la respuesta del jugador y la palabra correcta. Retorna la respuesta que dio el jugador y el puntaje (1 o -1 según si repondió bien o mal).

Lo que tienes que hacer

Chivelisión olvidó programar la función para encriptar/desencriptar según la clave `arquetipos`. Debes implementar este proceso en `def codificador(string)`, que recibe un `string` que tienes que encriptar o desencriptar¹. **Nota importante:** Esto **debe** ser realizado mediante programación funcional, pues el uso de esta técnica lleva puntaje.

Además, como puedes ver, la función `leer_archivo` solamente lee lo que el archivo contiene y no lo desencripta. Del mismo modo, la función `jugar` no verifica si la respuesta del jugador es “pasaprogra”. Para agregar las funcionalidades requeridas, deberás implementar los siguientes decoradores, ordenados según su dificultad.

- `@pasaprogra`: Este decorador debe verificar si la palabra ingresada como respuesta por el jugador en la función `jugar` es igual a “pasaprogra”. De ser así, debe retornar la respuesta y **cero** puntos. En cualquier otro caso, debe retornar lo que retorna la función decorada.
- `@esconder_palabra`: Este decorador debe modificar la oración del juego del ahorcado para incluir la palabra correcta **camuflada** con guiones bajos² en vez de “palabra”. Finalmente, debe entregar la palabra correcta y la oración con la respuesta camuflada.
- `@desencriptar(funcion_decodificadora)`: Este decorador debe poder desencriptar el contenido que genere `leer_archivo`. Debe recibir una función decodificadora y entregar los elementos desencriptados.
- `@encriptar(funcion_codificadora)`: Este decorador debe poder encriptar la respuesta entregada por el jugador en la función `jugar`. Debe recibir como parámetro una función codificadora y guardar la respuesta (encriptada) en un archivo llamado `resultados_encr.txt`.

Checklist

Después de haber programado todo, revisa que hayas decorado las funciones para lograr que el juego funcione:

- El jugador debe tener la capacidad de ingresar un **pasaprogra** en cualquiera de los dos juegos, para así saltarse la palabra.
- En el juego del ahorcado, las palabras deben estar **escondidas** (ej: c_s_)
- Para poder jugar ambas modalidades, la lectura de archivos debe ser **desencriptada**
- Cada vez que el jugador de una respuesta (correcta, incorrecta o pasaprogra), la respuesta debe ser **encriptada** y guardada en `resultados_encr.txt`.

¹ Recuerda que este sistema `arquetipos` tiene la bondad de que encriptar y desencriptar es lo mismo.

² **Ayuda:** La función `def camuflar(palabra)` (incluida) retorna `palabra` con la mitad de sus letras con guiones bajos.

Notas

- Solo puedes **agregar** líneas de código al archivo `main.py` si **borras o editas** una línea de código original tu nota automáticamente será un **1.0**

Requerimientos

- (1.80 pts) Encriptación de oraciones.
 - (0.80 pts) Correcta encriptación y desencriptación con la clave `arquetipos`.
 - (1.00 pts) Uso de programación funcional para encriptación y desencriptación.
- (1.00 pts) Decorador `@pasaprogra`
 - (0.50 pts) Verifica si la palabra es “pasaprogra”.
 - (0.50 pts) Retorna respuesta y puntaje correcto según el caso.
- (1.20 pts) Decorador `@esconder_palabra`
 - (0.40 pts) La palabra se camufla.
 - (0.40 pts) Se inserta la palabra camuflada en la posición respectiva de la oración.
 - (0.40 pts) Entrega la palabra correcta y la oración modificada.
- (0.80 pts) Decorador `@desencriptar(funcion_decodificadora)`.
 - (0.40 pts) El método desencripta un texto encriptado, dada una función decodificadora.
 - (0.40 pts) El método entrega el texto desencriptado.
- (0.80 pts) Decorador `@encriptar(funcion_codificadora)`.
 - (0.40 pts) El método puede encriptar si se le entrega una función codificadora.
 - (0.40 pts) Las respuestas encriptadas se guardan en un archivo.
- (0.40 pts) Aplicación de decoradores.
 - (0.10 pts) Lectura de archivos decorada correctamente.
 - (0.10 pts) Juego del ahorcado decorado correctamente.
 - (0.20 pts) Función `jugar` decorada correctamente.

Entrega

- **Lugar:** En su repositorio privado de GitHub, en la **carpeta** `Actividades/AC03/`
- **Hora del último *push*:** 16:40