Enunciado Trabajo Práctico Grupal

Objetivo

Escribir una aplicación en Python, que permita analizar y evaluar el diseño modular aplicado, a programas escritos en Python.

Descripción General

Se deberá tomar el conjunto de programas que formen la aplicación a analizar, y realizar una serie de operaciones, que permitan la generación de información, con el objetivo de evaluar si la construcción obedece a los paradigmas del diseño modular y la programación estructurada.

Los programas que forman parte de la aplicación a analizar, vendrán informados en un archivo, denominado, programas.txt. Cada línea en este archivo contendrá el camino completo al programa, incluido su nombre. En la primera línea del archivo, debe ir el programa en el que se encuentra el bloque principal de ejecución de la aplicación. Las líneas subsiguientes, contendrán aquellos programas que correspondan a librerías o módulos codificados por el usuario, y que también formen parte de la aplicación en cuestión.

Ejemplo de contenido de programas.txt
c:\\aplicacion\\programa_principal.py c:\\aplicacion\\m_ingresos.py c:\\aplicacion\\m_calculos.py c:\\aplicacion\\m_impresiones.py c:\\aplicacion\\m_generales.py

Los nombres y caminos utilizados en el cuadro, sólo son a modo de ejemplo; los reales podrán ser cualesquiera.

Descripción de las prestaciones

Al inicio, la aplicación deberá tomar el conjunto de programas que se encuentran en programas.txt y procesarlos de forma tal de obtener como salida dos archivos:

1. **fuente_unico.csv**: contendrá sólo código. Debe excluir todo tipo de comentario. Incluirá todas las funciones que forman parte de la aplicación, en orden alfabético.
   El primer campo será el nombre de la función y dará el orden al archivo alfabéticamente.
   El segundo campo serán los parámetros formales, incluidos los paréntesis.
   El tercer campo será el nombre del módulo al que pertenece la función.
   El resto de los campos, serán cada una de las instrucciones de código que forman parte de la función. Cada línea de código será un nuevo campo, por lo tanto se encontrarán separadas por una "," y la cantidad será variable, ya que dependerá de la cantidad de líneas de código que formen parte de la función.

2. **comentarios.csv**: contendrá sólo comentarios y el nombre de la función.
   El primer campo será el nombre de la función y dará el orden al archivo alfabéticamente.
   El segundo campo será el nombre del autor/responsable de la función. Para obtener esta información se debe extraer de la sección que se encuentra entre comillas triples e inmediatamente después de la firma de la función; lo que esté identificado con el siguiente marcado, a modo de ejemplo [Autor: Juan Perez].
   El tercer campo será la descripción de ayuda de uso de la función, que de igual modo será identificada, por ejemplo, [Ayuda: Obtener deuda del cliente en base a la tasa de actualización y totales adeudados recibidos]
   El resto de los campos, existirán, si hay algún otro tipo de comentario en la función. En este caso, cada comentario, aparecerá como un dato nuevo, separado por una "," de otro, si es que hay más de un comentario.

Para realizar el proceso de mezcla de los archivos, es necesario primero, hacer un proceso de reordenamiento en cada uno de ellos, que genere un archivo ordenado alfabéticamente para cada uno. Luego, se debe aplicar el algoritmo de mezcla dado en clase, para procesar secuencialmente todos los archivos ordenados, en forma paralela.

Una vez realizado el proceso mencionado y habiendo generado los archivos correspondientes, se deberá ofrecer un menú de opciones que permita acceder a las funcionalidades que se describen en los siguientes puntos. De aquí en más, todas las consultas deben realizarse sobre los archivos fuente_unico.csv y comentarios.csv, pudiendo almacenar en estructuras adecuadas, partes del archivo, en función de lo solicitado en cada punto.

1. Panel General de Funciones

Mediante esta opción se debe mostrar por pantalla, una tabla con la siguiente información por columna:

- *Nombre de la Función.Módulo* (**FUNCION**) – nombre función seguido de un punto y el nombre del módulo
- *Cantidad de Parámetros Formales* (**Parámetros**)
- *Cantidad de líneas de código* (**Líneas**) – no contabilizar las líneas vacías
- *Cantidad de invocaciones a la función* (**Invocaciones**)
- *Cantidad de Puntos de Salida* (**Returns**)
- *Cantidad de If*, contabilizar los anidamientos elif (**If/elif**)
- *Cantidad de For* (**for**)
- *Cantidad de While* (**while**)
- *Cantidad de Break* (**Break**)
- *Cantidad de Exit* (**Exit**)
- *Cantidad de líneas de comentarios* (**Coment**) – sólo los extra descripción
- *Indicador de Descripción Función* (**Ayuda**) – Si/No indicando que hay descripción de uso de la función, para lo cual se debe evaluar si inmediatamente después de la firma de la función, existe comentario entre comillas triples.
- *Autor/Responsable* (**Autor**)

Utilice como nombre de las columnas, los nombres que figuran entre paréntesis en cada caso.

Este punto también debe generar el archivo “panel_general.csv”, en el cual cada línea del archivo contenga la información descripta en cada uno de los puntos.
La primera línea del archivo, debe contener los nombres que figuran entre paréntesis.

2. **Consulta de Funciones**

Este punto deberá mostrar cada uno de los nombres de las funciones, ordenados alfabéticamente, uno al lado del otro, encolumnados; ocupando las líneas que sean necesarias; simulando una tabla como la que encontrará en el Built In Function de Python.org (https://docs.python.org/3/library/functions.html#compile)

Debajo de la tabla, debe aparecer el mensaje "Función: ", a la espera del ingreso de uno de los nombres listados seguido por alguno de los siguientes caracteres:

?	Indica que se debe mostrar la descripción asociada al uso de la función y que se encuentra entre comillas triples inmediatamente después del nombre de la función. También debe mostrar los parámetros formales que espera la función, el módulo al que pertenece, y el autor y/o responsable de la función.
#	Indica que se quiere ver todo lo relativo a la función.

Continuar solicitando el ingreso de nombres de funciones, hasta que el usuario sólo de enter.

En el caso que el usuario ingrese un nombre de una función inexistente y/o carácter inválido, enviar mensaje acorde y solicitar reingreso.

Si el usuario ingresa "**?todo**", debe listarse la información descripta, pero para cada una de las funciones por pantalla.
De igual modo, si ingresa "**#todo**".

Si el usuario ingresa "**imprimir ?todo**", debe enviarse al archivo ayuda_funciones.txt, el contenido correspondiente, formateado de forma tal que no supere 80 caracteres por líneas, y que la división entre función y función sea clara.

3. Analizador de Reutilización de Código

A través de este punto se debe reflejar mediante una tabla, quien invoca a quien/es, y quien es invocado por quien/es.

Ejemplo:

FUNCIONES	1	2	3	4	5	6	7
1 - main		1			1	1	1
2 obtener_datos	x		2				
3 - solicitar_valor		x		1		x	
4 - validar_valor			x				
5 - calcular_resultado	x						
6 - solicitar_rangos	x		1				
7 - imprimir_informe	x						
Total Invocaciones		1	3	3	1	1	1

Los valores representan la cantidad de veces que la función de la fila, invoca a la función de la columna. Por ejemplo, *solicitar_valor invoca 1 vez a validar_valor***.**

La "x" representa, la función de la fila, que es invocada por la función de la columna. Por ejemplo, *solicitar_valor es invocada por obtener_valor y por solicitar_rangos*.

El punto también debe generar el archivo "analizador.txt". Utilice caracteres en blanco para dar una disposición correcta a las "x" y los valores a mostrar, para que queden encolumnados adecuadamente.

4. Árbol de Invocación

Diagrama que gráfica la interacción entre las funciones, indicando quien llama a qué función.

```
main (6) --> ingresar_datos (8) --> solicitar_valor (5) --> validar_valor (5)
                                --> solicitar_valor (5) --> validar_valor (5)
         --> calcular_resultado (4)
         --> solicitar_rangos (5) --> solicitar_valor (5) --> validar_valor (5)
         --> imprimir_informe (7)
```

El valor que se encuentra entre paréntesis es la cantidad de líneas que tiene la función

5. Información por Desarrollador

Debe brindar datos sobre la participación de cada uno de los integrantes en el desarrollo de la aplicación. Para ello, deberá mostrar por pantalla, la siguiente información por Autor, ordenada en forma descendente, por líneas de código.

También deberá generar la misma salida al archivo “participacion.txt”

```
    Informe de Desarrollo Por Autor


Autor: Juan Perez

       Funcion                Lineas
       --------------------------------
       ingresar_datos           8
       solicitar_valor          5
       validar_valor            5
       solicitar_rangos         5
       4 Funciones - Lineas     23   57%


Autor: Ana Garcia

       Funcion                Lineas
       --------------------------------
       main                     6
       calcular_resultado       4
       imprimir_informe         7
       3 Funciones - Lineas     17   43%


Total: 7 Funciones - Lineas     40
```

Organización del Trabajo

Se sugiere seguir el siguiente esquema de organización:

1. **Cada integrante del equipo deberá leer detenidamente el enunciado**, y armar un listado con todas las dudas que le surjan sobre el enunciado. También deberá preparar un esquema preliminar para presentar a su grupo de como encararía la solución.

2. **El equipo debe establecer una primera reunión virtual** en la cual se tratarán de evacuar las dudas de los participantes del equipo, respecto del enunciado. Aquellas dudas que no puedan ser aclaradas entre los integrantes, serán consultadas a través del medio que se haya puesto a disposición para canalizarlas, y serán respondidas por los docentes, en los días de clase, en el espacio que destinen.
   En esta misma reunión, los integrantes presentarán sus esquemas de solución, y deberán arribar a uno consensuado. A partir de este, se dividirán los trabajos a realizar, estableciendo fechas de cumplimiento. Si alguno de los participantes no cumple con su parte, será decisión del equipo como proceder, teniendo en cuenta, que la falta o retraso de participación de uno ó más integrantes, no son justificativos para no entregar el producto completo.

3. Luego de la primera reunión, se debe establecer un nuevo esquema de reuniones y canales que permitan la interacción fluida del equipo.

4. **Deberán utilizar Github** como plataforma de desarrollo colaborativo, y en caso que el docente a cargo de la corrección del TP se los requiera, deberán darle acceso al proyecto.

5. A los fines de la entrega y la corrección del TP, **se tomará como aplicación a analizar, el conjunto de programas que formen parte del desarrollo realizado por ustedes**, ó sea los programas fuentes de la misma aplicación; pudiendo también, utilizarse **los fuentes de otros equipos**. Se sugiere, que inicialmente, creen un conjunto de programas de prueba, a los fines de facilitar el desarrollo de su aplicación.

Condiciones de Entrega

La entrega del presente trabajo práctico está a definir, pero será aproximadamente la semana del 21 de Julio y deberán subirlo al campus mediante la tarea que se encontrará habilitada a tal fin.

El presente enunciado podrá sufrir modificaciones, y en tal caso serán informadas oportunamente.